N° d'Ordre :



THÈSE

présentée devant

L'UNIVERSITÉ DE LA NOUVELLE-CALÉDONIE

et

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

pour obtenir

LE GRADE DE DOCTEUR

Spécialité

INFORMATIQUE

Ecole Doctorale Pluridisciplinaire Numérique
des Milieux Insulaires Ultra-Marins

présentée par

Dominique Joël GAY

CALCUL DE MOTIFS SOUS CONTRAINTES POUR LA CLASSIFICATION SUPERVISÉE

Soutenue publiquement le 30 novembre 2009 devant le jury :

Henri Bonnel, Professeur, Université de la Nouvelle-Calédonie	Président
Bruno Crémilleux, Professeur, Université de Caen	Rapporteur
Ho Tu-Bao, Professeur, Japan Advanced Institute of Science and Technology	Rapporteur
Marc Boullé, Chercheur, France Télécom R&D	Examinateur
Eibe Frank, Associate Professor, Université de Waikato	Examinateur
Nazha Selmaoui-Folcher, Maître de Conférences, Université de la Nouvelle Calédonie	Co-directeur de thèse
Jean-François Boulicaut, Professeur, INSA-Lyon	Co-directeur de thèse

Remerciements

Tout d'abord, je tiens à remercier mes directeurs de thèse Nazha Selmaoui-Folcher et Jean-François Boulicaut pour leur encadrement, leur aide, leurs conseils et encouragements tout au long de ce travail.

Je souhaite également remercier Bruno Crémilleux et Ho Tu-Bao pour m'avoir fait l'honneur d'accepter d'être rapporteurs de ce mémoire de thèse. Je remercie également Marc Boullé, Eibe Frank et Henri Bonnel pour avoir participé à l'évaluation de mon travail lors de la soutenance du mémoire.

Je remercie aussi les membres passés et présents des équipes PPME et ERIM de l'Université de la Nouvelle-Calédonie et de l'équipe TURING du LIRIS à l'INSA-Lyon pour leur accueil chaleureux. Merci en particulier à Christophe Rigotti, Claire Leschi, Jérémy Besson et Frédéric Flouvat pour leurs discussions porteuses d'idées. Merci à Isabelle Rouet pour sa collaboration et son courage ... il en faut pour expliquer à des informaticiens certaines facettes de l'érosion des sols. Merci aussi à Loïc Cerf pour ses discussions passionnées sur certains aspects de la fouille de données mais aussi sur d'autres sujets de *geek*.

Je remercie bien sûr ma famille et mes amis pour m'avoir soutenu pendant toute la durée de ce travail.

Enfin, merci à Virginie pour sa présence en chaque instant.

A Jean-Marc
et Maïdhili

~

On se réjouissait à ta naissance et tu pleurais.
Vis de manière que tu puisses te réjouir au moment de ta mort
et voir pleurer les autres.
Proverbe persan

Résumé

Ces dernières années, l'extraction de motifs locaux (itemsets fréquents et règles d'association) a suscité beaucoup d'entrain pour la classification supervisée. Cette thèse traite du calcul et de l'usage de motifs sous contraintes pour la classification supervisée. Nous nous attaquons à deux problèmes difficiles en classification supervisée à base de motifs et proposons deux contributions méthodologiques :

- *(i)* D'un côté, lorsque les attributs sont bruités, les performances des classifieurs peuvent être désastreuses. Les méthodes existantes consistent à corriger les valeurs d'attributs ou supprimer les objets bruités – ce qui génère une perte d'information. Dans ce mémoire, nous proposons une méthode générique de construction de descripteurs robustes au bruit d'attributs – sans modifier les valeurs d'attributs ni supprimer les objets bruités. Notre approche se déroule en deux étapes : premièrement nous extrayons l'ensemble des règles δ-fortes de caractérisation. Ces règles offrent des propriétés de corps minimal, de non-redondance et sont basées sur les itemsets δ-libres et leur δ-fermeture – qui ont déjà fait leur preuve pour la caractérisation de groupements dans des contextes bruités. Deuxièmement, nous construisons un nouveau descripteur numérique robuste pour chaque règle extraite. Les expérimentations menées dans des données bruitées, montrent que des classifieurs classiques sont plus performants en terme de précision sur les données munies des nouveaux descripteurs que sur les données avec les attributs originaux.
- *(ii)* D'autre part, lorsque la distribution des classes est inégale, les approches existantes de classification à base de motifs ont tendance à être biaisées vers la classe majoritaire. La précision sur la (ou les) classe(s) majoritaire(s) est alors élevée au détriment de la précision sur la (ou les) classe(s) minoritaire(s). Nous montrons que ce problème est dû au fait que les approches existantes ne tiennent pas compte de la répartition des classes et/ou de la fréquence relative des motifs dans chacune des classes de la base. Pour pallier ce problème, nous proposons un nouveau cadre de travail dans lequel nous extrayons un nouveau type de motifs : les règles de caractérisation `One-Versus-Each` (`OVE`-règles). Ce nouveau cadre de travail nécessite le paramétrage d'un nombre conséquent de seuils de fréquence et d'infréquence. Pour ce faire, nous proposons un algorithme d'optimisation de paramètres, `fitcare` ainsi qu'un algorithme d'extraction d'`OVE`-règles. Les expérimentations menées sur des données `UCI` multi-classes disproportionnées et sur des données de diagnostic de méningite aiguë, montrent que notre approche `fitcare` est plus performante que les approches existantes en terme de précision sur les classes mineures.

L'application de notre méthode de classification associative à l'analyse de données d'érosion des sols en Nouvelle-Calédonie a mis en évidence l'intérêt de notre proposition pour caractériser les phénomènes d'érosion.

Mots-clés : Extraction de motifs sous contraintes, Classification Associative, Construction de Descripteurs, Tolérance au Bruit, Problèmes Multi-Classes inégalement distribuées

Abstract

Recent advances in local pattern mining (eg. frequent itemsets or association rules) has shown to be very useful for classification tasks. This thesis deals with local constraint-based pattern mining and its use in classification problems. We suggest methodological contributions for two difficult classification tasks :

- *(i)* When training classifiers, the presence of attribute-noise can severely harm their performance. Existing methods try to correct noisy attribute values or delete noisy objects – thus leading to some information loss. In this thesis, we propose an application-independent method for noise-tolerant feature construction – without modifying attribute values or deleting any objects. Our approach is two-step : Firstly, we mine a set δ-strong characterization rules. These rules own fair properties such as a minimal body, redundancy-awareness and are based on δ-freeness and δ-closedness – both have already served as a basis for a fault-tolerant pattern and for cluster characterization in noisy data sets. Secondly, from each extracted rule, we build a new numeric robust descriptor. The experiments we led in noisy environments have shown that classical classifiers are more accurate on data sets with the new robust features than on original data – thus validating our approach.
- *(ii)* When class distribution is imbalanced, existing pattern-based classification methods show a bias towards the majority class. In this case, accuracy results for the majority class are abnormally high to the expense of poor accuracy results for the minority class(es). In this thesis, we explain the whys and whens of this bias. Existing methods do not take into account the class distribution or the error repartition of mined patterns in the different classes. In order to overcome this problem, we suggest a new framework and deal with a new pattern type to be mined : the `One-Versus-Each`-characterization rules (`OVE`). However, in this new framework, several frequency and infrequency thresholds have to be tuned. Therefore, we suggest `fitcare` an optimization algorithm for automatic parameter tuning in addition to an extraction algorithm for `OVE`-characterization rule mining. The experimentations on imbalanced multi-class data sets have shown that `fitcare` is significantly more accurate on minor class prediction than existing approaches.

The application of our `OVE` framework to a soil erosion data analysis scenario has shown the added-value of our proposal by providing a soil erosion characterization validated by domain experts.

Keywords : Constraint-based Pattern Mining, Pattern-based classification, Feature Construction, Noise-Tolerance Classification, Imbalanced Data Sets, Multi-class Classification

Note for English readers : To enjoy the main contributions of this thesis, English readers may refer to [GSB08, CGSB08, GSB09].

Table des matières

Table des figures

Liste des Algorithmes

Première partie

Introduction

Introduction

Ce manuscrit présente nos travaux de recherche sur l'exploitation de motifs dans des processus de classification supervisée. Le domaine d'application choisi est l'analyse des phénomènes d'érosion des sols. Les principales contributions sont d'ordre méthodologique. Tout d'abord, nous proposons des méthodes génériques, c'est-à-dire indépendantes d'un domaine d'application particulier, pour calculer des motifs locaux utiles à la construction de modèles prédictifs (classification supervisée). Nous nous intéressons ensuite à des contextes de classification réputés difficiles comme, par exemple, la construction de descripteurs robustes lorsque les exemples d'apprentissage sont bruités ou encore le cas des répartitions de classes possiblement nombreuses et déséquilibrées.

Ce doctorat a été préparé sous la tutelle de deux universités – l'Université de la Nouvelle-Calédonie (UNC) et l'Institut National des Sciences Appliquées de Lyon (INSA-Lyon) – et nos travaux ont été réalisés au sein des équipes ERIM EA3791 et PPME EA3325 à l'UNC et de l'équipe TURING du LIRIS-CNRS UMR5205 à l'INSA-Lyon.

Contexte

Les équipes TURING et l'équipe "Data Mining" des EA ERIM/PPME ont pour axe de recherche commun la "fouille de données". Une partie des efforts de recherche est dédiée à l'extraction de motifs dans les données Booléennes (dans la littérature, elles sont aussi appelées données transactionnelles) et aux usages multiples de ces motifs.

La fouille de données est une partie intégrante du processus de découverte de connaissances dans les bases de données (en anglais KDD pour Knowledge Discovery in Databases). La communauté internationale de fouille de données s'accorde sur les principales étapes du processus KDD [FPSSU96, HK00, TSK05]. Nous rappelons les différentes étapes de ce processus en figure 1.

La première étape de pré-traitement consiste à transformer les données brutes en un format approprié pour les étapes suivantes d'analyse. Des exemples de pré-traitement

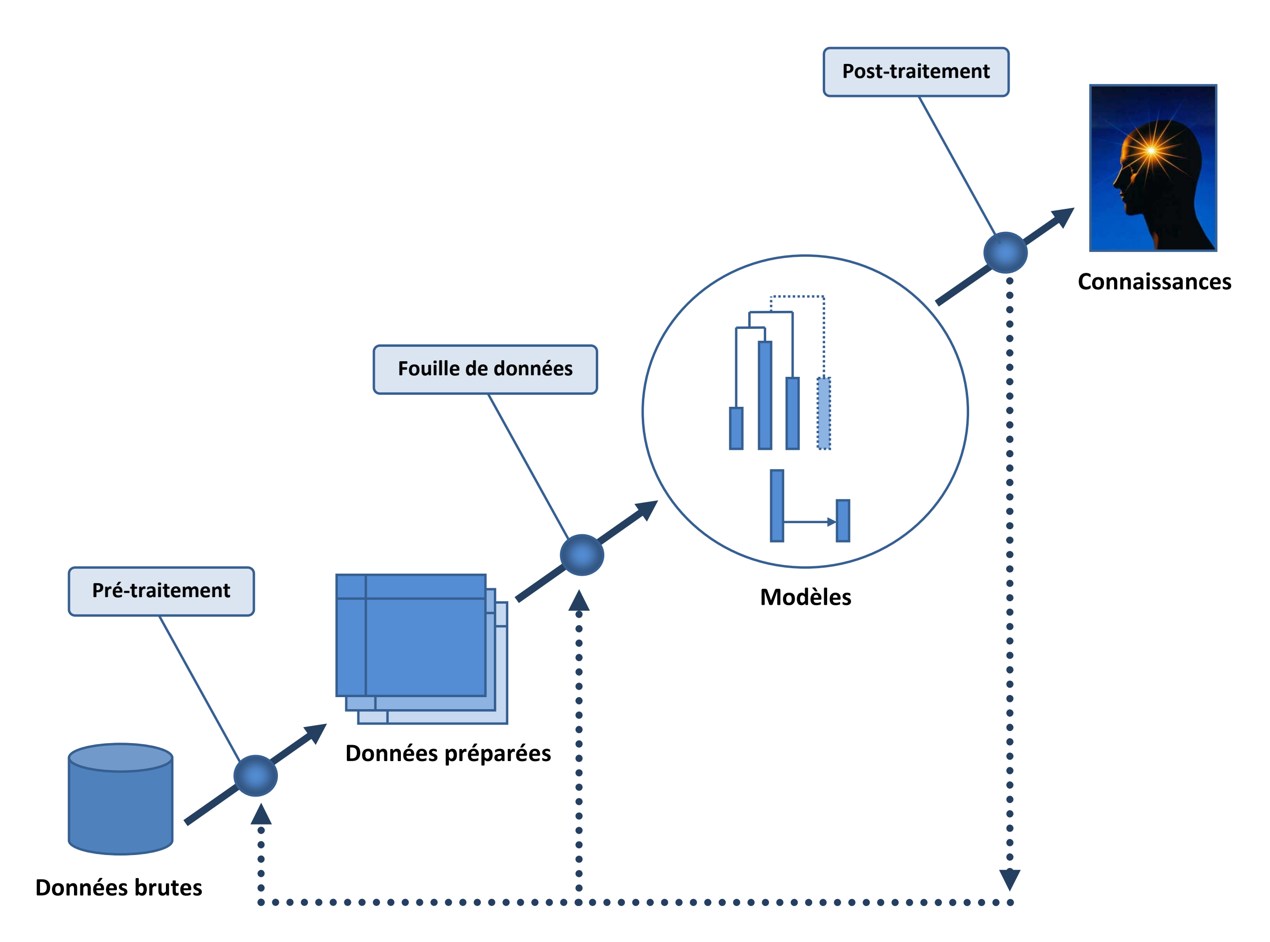


Figure 1 – Processus d'extraction de connaissances dans les données

sont la sélection d'un sous-ensemble des enregistrements, la sélection de sous-ensembles d'attributs descripteurs, la construction de nouveaux attributs descripteurs, une normalisation de certains attributs, la discrétisation ou binarisation des attributs numériques, l'élimination du bruit dans les données, le traitement des valeurs manquantes, etc. Il faut également travailler ici à la définition des éventuels paramètres d'entrée de la tâche de fouille de données à réaliser.

Selon [TSK05], les tâches de fouille de données peuvent être considérées selon deux catégories :

- Les tâches descriptives : le but est d'extraire des motifs (e.g. des corrélations entre ensemble d'attributs, des tendances dans les données, des groupes pertinents d'enregistrements ou clusters ou encore des anomalies dans les données, . . .) qui résument les relations sous-jacentes aux données.
- Les tâches prédictives : le but est de prédire la valeur d'un attribut particulier à l'aide des valeurs des autres attributs. Cet attribut particulier est souvent appelé attribut classe ou label tandis que les autres attributs sont appelés des descripteurs.

Le post-traitement peut consister en la visualisation, l'interprétation ou l'évaluation

des résultats de la fouille. Mais aussi, cette étape peut avoir pour but d'intégrer les modèles construits (descriptifs ou prédictifs) dans des systèmes d'aide à la décision.

Notez que le processus KDD se veut itératif. Ainsi, l'interprétation des informations ou connaissances obtenues à l'étape de post-traitement peuvent nous conduire à réitérer tout ou partie du processus en utilisant ces mêmes connaissances selon les directives des experts du domaine d'application.

Dans ce manuscrit, nous nous focalisons sur la classification supervisée pour les tâches de prédiction appliquées à des données binaires. Les données d'entrée pour un algorithme de classification supervisée sont des enregistrements (également appelés exemples). Chaque enregistrement sera pour nous un tuple (I, c), où I est un ensemble d'attributs et c un attribut particulier, l'attribut cible (ou classe). Nous nous restreignons au cas où c est attribut nominal. Nous ne parlerons donc pas des méthodes de regression qui s'appliquent lorsque l'attribut classe est de type numérique. Le but d'un processus de classification supervisée est d'apprendre une fonction surjective qui à chaque enregistrement associe une valeur de l'attribut classe c. La fonction apprise est appelée modèle de classification (ou classifieur). Ce modèle de classification peut être utilisé par la suite pour la phase de prédiction où l'on assigne une classe à de nouveaux enregistrements entrants (voir figure 2).

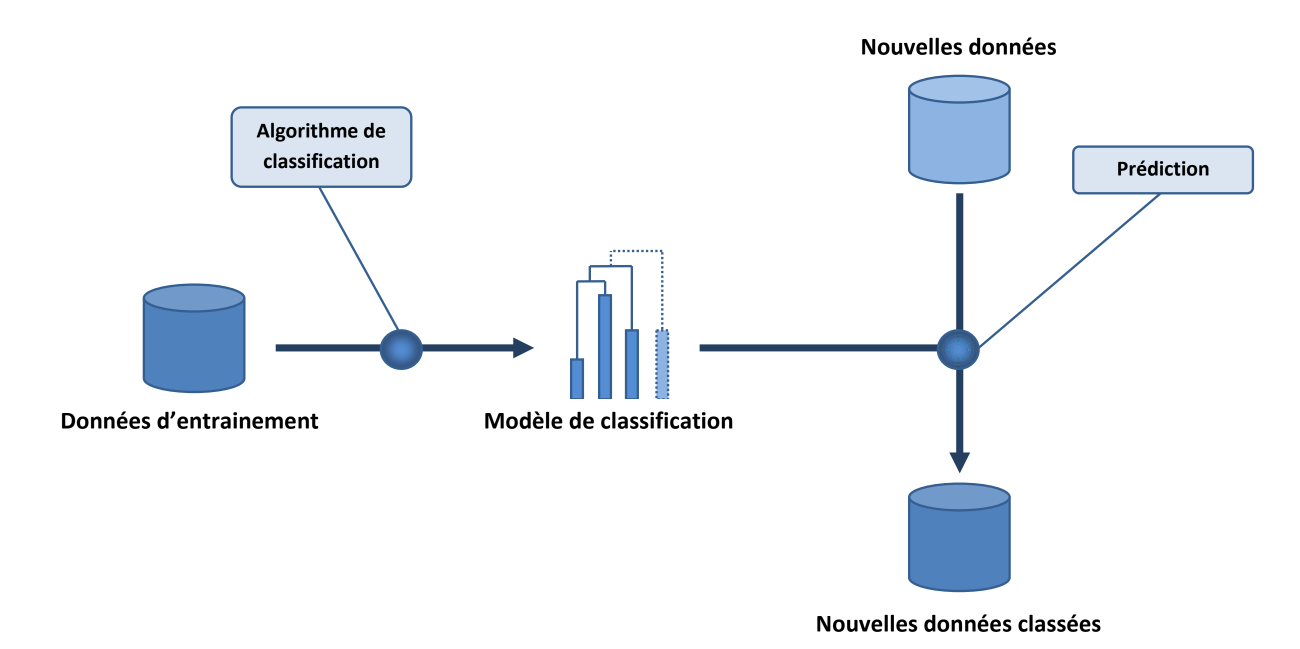


FIGURE 2 – Processus de classification supervisée et prédiction

Classification supervisée à base de motifs locaux

Les domaines de l'apprentissage automatique et des statistiques ont donné lieu à une multitude de méthodes de classification supervisée. Parmi les plus connues, on trouve la construction d'arbres de décision, celle de classifieurs bayésiens, l'induction de règles de classification, l'apprentissage de réseaux de neurones ou encore de "Support Vector Machines" ou SVM (appelés Séparateurs à Vastes Marges dans [CM02]). Nous renvoyons le lecteur, par exemple à [HK00], pour une étude approfondie.

D'autre part, depuis le début des années 90, de nombreux chercheurs se sont intéressés à la tâche descriptive de l'extraction de motifs fréquents. Cet effort de recherche a motivé l'étude de méthodes de classification supervisée qui produiraient des classifieurs ex exploitant de tels motifs. Dans des données transactionnelles, les motifs fréquents sont typiquement des sous-ensembles d'attributs (itemsets) dont le nombre d'occurrences est significatif, i.e. supérieur à un seuil donné. L'ensemble des itemsets fréquents capture donc certaines tendances ou régularités dans les données. L'intuition serait que *"Ce qui est fréquent peut être intéressant"*. En classification supervisée, ce que nous recherchons, ce sont des mécanismes qui sont discriminants pour l'attribut classe. Ainsi, un motif sera intéressant s'il sépare bien les objets qui le "respectent" de ceux qui ne le "respectent" pas au regard des étiquettes de classes disponibles. La figure 3 confirme bien cette intuition. Dans les deux graphiques, pour la base de données `UCI wine` [AN07], nous représentons la valeur de gain d'information pour chaque itemset en fonction, respectivement, de leur taille k et de leur fréquence. Plus le gain d'information pour un motif est grand, plus ce motif est discriminant. Nous voyons clairement que les plus grandes valeurs de gain d'information sont atteintes pour des valeurs de $k \neq 1$ et des valeurs de fréquences différentes des extrêmes. Ainsi, il serait dommage de se limiter aux attributs simples car cela équivaudrait à se priver du potentiel de discrimination des k-itemsets ou ensembles de taille k. De plus, les itemsets très peu fréquents ont une valeur de gain d'information limitée. Les itemsets fréquents peuvent donc être utiles pour un processus de classification supervisée.

L'une des applications les plus étudiées des itemsets fréquents est la découverte de règles d'association, un problème introduit dans [AIS93]. Le but est d'extraire l'ensemble des règles de la forme $\pi : I \rightarrow J$ (où I l'antécédent et J le conséquent sont des ensembles d'attributs ou itemsets disjoints) qui satisfont certaines contraintes d'intérêt – par exemple, selon des seuils de valeurs pour une mesure d'intérêt donnée. Les premières études se sont focalisés sur les règles d'association valides, i.e., respectant une contrainte de fréquence minimale et une contrainte de confiance minimale. La fréquence, notée $freq(\pi, r)$ est le nombre d'occurrences de π dans les données r. La confiance est la probabilité conditionnelle (dans les données) que tous les attributs du conséquent d'une règle appartiennent à une transaction qui implique tous les attributs de l'antécédent – soit $freq(\pi, r)/freq(I, r)$. Il existe d'autres mesures d'intérêt dans la littérature ; la plupart de ces mesures étant souvent basées sur la notion de fréquence.

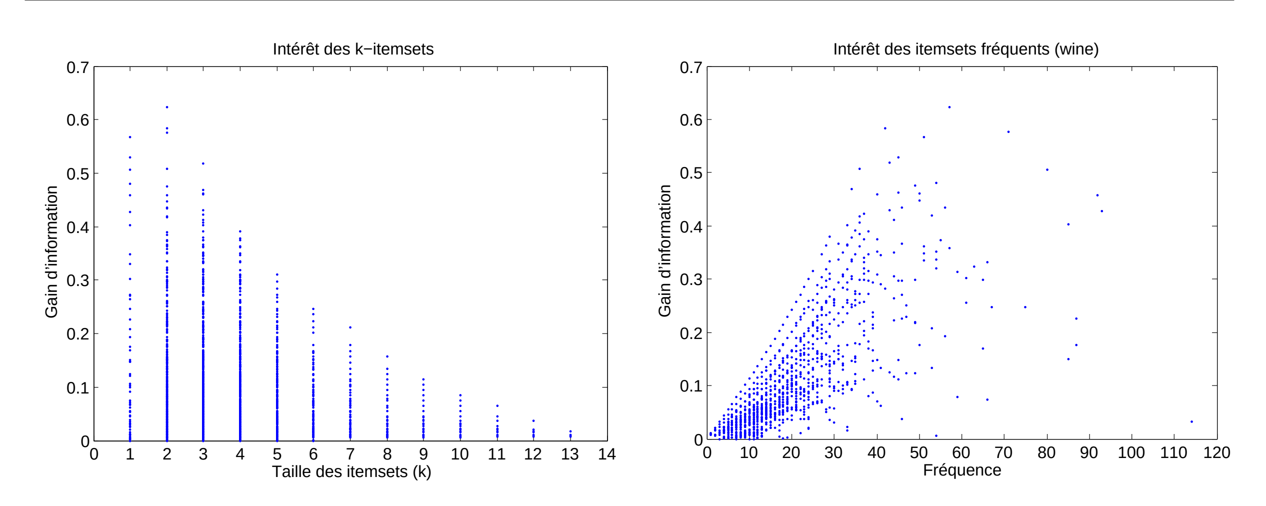


FIGURE 3 – Gain d'information des itemsets en fonction de leur taille et de leur fréquence pour les données `wine`

Bien qu'il soit important de différencier l'extraction de règles d'association du calcul de règles pour la classification supervisée [Fre00], on sent bien que, sous certaines conditions, une règle d'association qui conclut sur un attribut classe peut être utile pour construire un modèle de classification. Les pionniers en la matière [LHM98] se sont donc intéressés à l'ensemble des règles d'association concluant sur un attribut classe et respectant des seuils de fréquence et de confiance minimum. L'ensemble des règles extraites est ensuite ordonnée sous forme de liste selon leur valeur de confiance et de fréquence. Cet ensemble ordonné forme le modèle de classification. Ainsi, la première règle de la liste supportée par un nouvel exemple entrant t indique la classe à prédire pour t.

Depuis, d'autres classifieurs associatifs ont été développés [LHP01, YH03, WK05]. Nous reviendrons en détails sur ces méthodes dans la seconde partie. Bien qu'il existe différentes méthodes de classification supervisée à base de motifs, les différents auteurs s'accordent sur plusieurs points clés que se doit de respecter l'ensemble de motifs afin d'espérer de bonnes performances de classification :

- *(i)* Les motifs de l'ensemble doivent être intéressants selon une mesure d'intérêt discriminante pour l'attribut classe.
- *(ii)* L'ensemble des motifs doit offrir une bonne couverture des données d'apprentissage, i.e., il faut que la quasi-totalité des transactions utilisées lors de l'apprentissage puissent être couvertes par au moins un motif.
- *(iii)* L'ensemble des motifs doit être concis et sans redondance.

Problèmes identifiés

Le contexte étant posé, nous identifions quelques problèmes ouverts en classification supervisée à base de motifs.

Représentations condensées pour la classification supervisée

La principale faiblesse commune aux approches de classification basées sur les motifs (itemsets ou règles d'associations) est le grand nombre de motifs extraits. En effet, pour capturer certaines tendances dans les données, un seuil de fréquence assez bas peut être requis. On se retrouve alors devant un très grand nombre de motifs fréquents à extraire. De plus, l'ensemble résultat peut contenir et des motifs redondants et des motifs inutiles lorsqu'on est face à des données imparfaites au sens où elles pourraient être bruitées. Dans de tels cas, les temps d'extraction sont très longs et les phases de post-traitement deviennent difficiles et très couteuses.

Un progrès essentiel pour la faisabilité et la pertinence des calculs de motifs fréquents réside dans l'étude des représentations condensées, notamment celles qui exploitent des propriétés de fermetures [BB00, BTP+00, Zak00]. Intuitivement, une représentation condensée d'une collection de motifs fréquents est une représentation alternative et plus concise permettant, si besoin est, de retrouver tous les motifs fréquents et leurs fréquences sans avoir à accéder aux données [MT96]. Le concept est général et peut être étudié pour différents types de motifs et différentes mesures, i.e., pas seulement la mesure de fréquence. Les avantages des représentations condensées sont clairs : elles peuvent être extraites plus efficacement (en terme de temps et d'espace) que les collections de motifs qu'elles permettent de retrouver.

La formalisation de [BTP+00] est élégante. Les auteurs proposent de grouper les itemsets ayant le même support et supportés par un même ensemble de transactions. Ainsi, les classes d'équivalence de support contiennent des itemsets qui ont le même support et donc la même fréquence. Avec un itemset par classe d'équivalence et sa fréquence, il est donc possible de déduire la valeur de fréquence de tous les autres itemsets de la classe d'équivalence. Cette possibilité d'inférer la fréquence des itemsets améliore considérablement les performances d'extraction. Plusieurs représentations condensées basées sur ces idées ont été proposées. Elles peuvent être basées sur des itemsets particuliers des classes d'équivalence comme, par exemple, les itemsets fermés – l'unique maximal (au sens de l'inclusion) d'une classe d'équivalence ou bien les itemsets clés ou libres – les minimaux d'une classe d'équivalence. Nous renvoyons le lecteur à [CRB05] pour une synthèse sur cette question.

Puisque les itemsets d'une même classe d'équivalence ont la même fréquence, ils ont

aussi la même valeur d'intérêt pour toute mesure d'intérêt basée sur la fréquence. Pour éviter de la redondance, un seul itemset par classe d'équivalence peut être retenu. Le choix d'un tel itemset représentant pour la classification supervisée a été beaucoup discuté. Certains auteurs suggèrent les itemsets fermés [GKL06, CYHH07], d'autres les itemsets libres [BC04, LLW07]. Il nous a semblé important de mieux comprendre les argumentaires des uns et des autres et d'apporter à notre tour quelques éléments de réponse sur cette question (voir chapitre 3).

Classification supervisée dans les données bruitées

Il est admis que les données d'entrée sont rarement parfaites. Souvent, la collection des données reste problématique et imprécise, les étapes de discrétisation peuvent produire des codages Booléens regrettables ou encore l'étiquetage des données d'apprentissage (valeur de l'attribut classe) sera sujet à caution. En classification supervisée, la présence de bruit peut avoir un impact négatif sur la performance des classifieurs et, par conséquent, sur la pertinence des décisions prises avec ces modèles [ZW04]. On peut identifier deux types de bruits dans des données binaires : le bruit de classe, lorsque le bruit affecte l'attribut classe ; et le bruit d'attributs lorsque le bruit affecte uniquement les attributs non-classe. Le bruit de classe a été intensivement étudié dans la littérature. Le problème du bruit d'attribut reste insuffisamment étudié. Si les méthodes de traitement du bruit de classe suivant le processus "détection - correction/délétion" améliorent la performance des classifieurs, rien n'est garanti lorsqu'on est face au bruit d'attribut. En effet, corriger les valeurs des attributs soi-disant détectés ne nous rend pas des données parfaites et supprimer les attributs ou transactions bruités peut conduire à une perte inacceptable d'information. Ainsi, la performance des classifieurs est détériorée.

Dans ce manuscrit, nous nous intéressons au problème de la classification supervisée basée sur les motifs en présence de bruit d'attribut. Dans un tel contexte, le nombre d'itemsets fermés explose car les motifs fermés que nous devrions avoir en l'absence de bruit deviennent fragmentés. De fait, les motifs fermés qui sont alors retrouvés ne sont plus assez représentatifs des tendances qu'il faudrait retrouver dans les données. Ce problème motive les travaux récents sur la détection de motifs qui tolèrent des exceptions et, notamment, des extensions du concept d'itemset fermés pour une tolérance aux erreurs (voir, par exemple, [BRB06] pour une proposition, ou encore [GFF$^+$08] pour une synthèse). Dans [BBR00, BBR03], les auteurs proposent une représentation condensée approximative basée sur les itemsets δ-libres. Dans cette approche, l'approximation est gouvernée par un entier δ qui indique un nombre d'exceptions maximal par attribut. Ainsi, au lieu de regrouper les itemsets qui ont un support équivalent, on regroupe les itemsets ayant presque le même support (avec un gap maximum de δ entre les supports des différents itemsets). Cette généralisation a été créée pour pouvoir approximer la valeur de la fréquence des autres itemsets (non-δ-libres) dans des contextes difficiles. Cependant, les itemsets δ-

libres et leur δ-fermeture ont déjà été utilisés pour différentes tâches de fouille de données comme l'extraction d'itemsets fréquents, la découverte de sous-ensembles de règles d'association a priori intéressantes ou encore la caractérisation de clusters. Dans le chapitre 3, nous considérons leurs utilisations dans un contexte de classification supervisée en proposant une méthode générique de construction de nouveaux descripteurs tolérants aux bruits d'attributs.

Problèmes multi-classes inégalement distribuées

Le problème de la fouille de données sur des classes inégalement distribuées est l'un des dix problèmes ouverts mis en avant par la communauté scientifique [1]. Dans ce manuscrit, nous proposons des éléments de solution pour la mise en oeuvre d'une classification supervisée basée sur des motifs dans les bases de données multi-classes et inégalement distribuées. De manière informelle, dans ce type de problème, le nombre de classes est supérieur à deux et parmi celles-ci au moins une (appelée classe minoritaire) comporte beaucoup moins d'objets que certaines autres (classes majoritaires). Dans ce contexte, les approches utilisant le cadre fréquence-confiance atteignent leurs limites. En effet, pour espérer capturer des motifs qui caractérisent une classe minoritaire, le seuil de fréquence doit être bas (bien inférieur à la taille de la classe minoritaire). Imposer un tel seuil global peut générer des motifs inintéressants pour la classe majoritaire. De plus, il a été montré que le cadre fréquence-confiance est biaisé vers la classe majoritaire [VC07]. Ce biais vient du fait que la taille des différentes classes n'est pas prise en compte dans un cadre fréquence-confiance. Les performances des classifieurs sont alors détériorées, en particulier pour les classes minoritaires.

Dans [DL99], les auteurs proposent un principe de classification basé sur le concept de motif émergent. Les motifs émergents sont les motifs qui sont significativement plus fréquents dans une partie des données que dans le reste de la base. La mesure d'intérêt qui caractérise les motifs émergents est le taux d'accroissement. Le taux d'accroissement d'un motif I pour une classe c_i est simplement le rapport entre la fréquence relative de I dans r_{c_i} et la fréquence relative de I dans le reste des données $r \setminus r_{c_i}$: soit $(freq(I, r_{c_i})/|r_{c_i}|)/(freq(I, r \setminus r_{c_i})/|r \setminus r_{c_i}|)$. La construction de classifieurs basés sur des motifs émergents a bien été étudiée [DZWL99, LDR00b, LDR00a, LDR01, LRD01] et ces modèles ont fait leurs preuves. Cependant, les motifs émergents d'une classe c_i tiennent compte de la taille c_i et de la taille du reste des données. Malheureusement, si le reste des données est composée de plusieurs classes, les approches actuelles à base de motifs émergents n'en tiennent pas compte. Ainsi, dans certains cas, un motif émergent pour une classe c_i pourra tout aussi bien être émergent pour une autre classe c_j appartenant au reste des données. Le résultat sera l'apparition de conflits de motifs et donc une dégradation de la performance des classifieurs utilisés.

1. 10 challenging problems at http://www.cs.uvm.edu/~icdm/

D'une manière générale, les classifieurs à base de motifs existants suivent une approche dite OVA (One Versus All), i.e., pour un motif donné, on s'intéresse à sa fréquence dans une classe donnée et à sa fréquence dans le reste des données. Nous pensons que la nature même de cette approche est la raison première des problèmes rencontrés par les classifieurs à base de motifs dans les données multi-classes inégalement distribuées – en particulier en ce qui concerne la faible précision dans les classes minoritaires et le biais des classifieurs OVA vers la classe majoritaire. Dans le chapitre 4, nous mettons en évidence les problèmes rencontrés par les approches OVA et proposons une nouvelle méthode pour pallier aux problèmes identifiés. Plus précisément, nous proposons un classifieur à base de motifs spécialement dédié à ce type de problème en suivant une approche dite OVE (One Versus Each) où pour un motif donné, on s'intéressera à sa fréquence dans une classe c_i et à sa fréquence dans chacune des autres classes $c_j (j \neq i)$ du reste des données.

Organisation du mémoire

Ce mémoire est organisé en cinq parties de la manière suivante :

La prochaine partie est consacrée à un état de l'art de nos deux thèmes centraux. Le chapitre 1 passe en revue les principales approches de classification supervisée à base de motifs. Nous rappellerons les méthodes à base de règles inductives, de règles association, et d'itemsets fréquents. Dans le chapitre 2, nous exposons l'existant en matière de représentations condensées des itemsets fréquents ainsi que leurs usages multiples en fouille de données. Dans cette partie, nous poserons aussi le cadre théorique de l'extraction de motifs sous contraintes [BRM05] ainsi que les définitions nécessaires aux développements de nos contributions.

La troisième partie décrit nos deux principales contributions à la classification supervisée à base de motifs dans des contextes difficiles. Le chapitre 3 présente notre méthode de construction de descripteurs basée sur les itemsets δ-libres pour la classification supervisée de données binaires éventuellement bruitées. Puis, dans le chapitre 4, nous développons une nouvelle méthode de classification associative dédiée aux problèmes multi-classes inégalement distribuées. Ces deux chapitres contiennent également les indispensables études expérimentales qui permettent l'étude empirique de nos propositions.

En quatrième partie, nous développons un scénario d'extraction de connaissance pour l'analyse de l'érosion des sols en Nouvelle-Calédonie.

Enfin, la cinquième partie propose un bilan des travaux menés au cours de cette thèse et ouvre sur des perspectives de travaux futurs.

Deuxième partie

Etat de l'art

Chapitre 1

Usage multiple des motifs locaux en classification supervisée

Sommaire

1.1 Contexte général

Dans ce chapitre, nous donnons le contexte de travail, i.e. les bases de données transactionnelles binaires labellisées puis nous passons en revue les principales approches de classification supervisée basée sur les motifs locaux (itemsets ou règles).

Définition 1 (Base de données transactionnelles binaires labellisées) *Une base de données binaires (ou contexte binaire) est un triplet* $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ *où* $\mathcal{T}$ *est un ensemble d'objets appelés aussi transactions,* $\mathcal{I}$ *un ensemble d'attributs Booléens appelés aussi items ou propriétés et* $\mathcal{R}$ *une application telle que* $\mathcal{R} : \mathcal{T} \times \mathcal{I} \mapsto \{0, 1\}$*. Lorsque* $\mathcal{R}(t, i) = 1$*, on dit que la transaction* t *contient l'item* i *– ou encore l'objet* t *respecte la propriété* i*. On distingue les attributs classe (ou labels) des autres attributs :* $\mathcal{C} = \{c_1, c_2, \ldots, c_p\} \subseteq \mathcal{I}$*.*

Utiliser les motifs locaux pour la classification supervisée semble intuitif. Bien qu'il existe différentes approches, le processus utilisé est générique (cf figure 1.1) : *(i)* à partir des données binaires, on extrait un ensemble de motifs, puis *(ii)* à partir de l'ensemble de motifs extraits, on construit un classifieur.

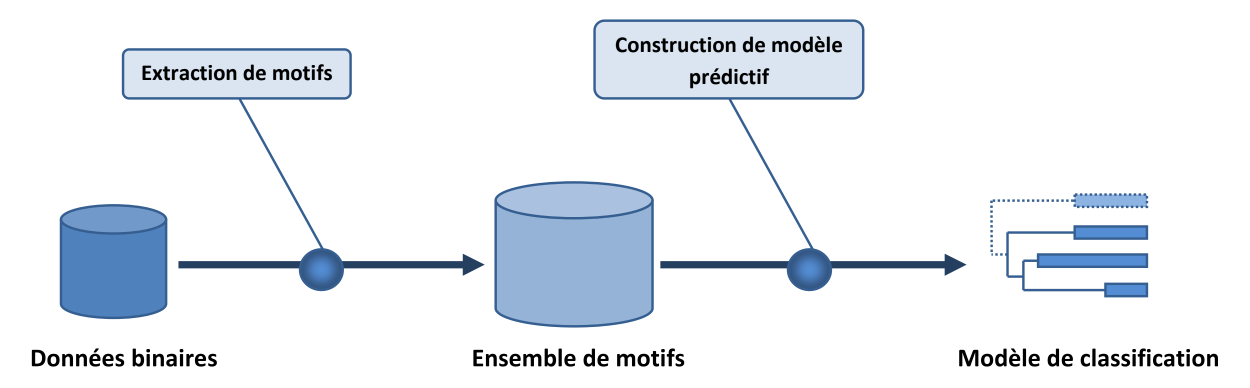


FIGURE 1.1 – Processus de classification supervisée à base de motifs

Si la manière de combiner les motifs locaux (afin de construire un classifieur) est importante et reste un problème ouvert, l'extraction de motifs est l'autre phase critique du processus. En effet, la qualité (en terme de précision) d'un classifieur à base de motifs dépend fortement de la qualité de l'ensemble des motifs extraits. Dans la littérature, les auteurs s'accordent sur certains points-clés qui caractérisent un "bon" ensemble de motifs pour la classification supervisée : chaque motif doit être considéré comme intéressant par rapport à une mesure d'intérêt ; pour être représentatif des données d'apprentissage, l'ensemble des motifs extraits doit couvrir une grande majorité des objets ; enfin l'ensemble des motifs doit être concis et sans redondance. Dans la suite, nous ferons le lien entre chacun de ces points-clés et chacune des méthodes exposées. Toutes les méthodes présentées dans ce chapitre sont de type `OVA` (One-versus-All) : les motifs extraits sont caractéristiques d'une classe par rapport à l'union des autres classes.

1.2 Méthodes à base de règles

Les méthodes de classification supervisée à base de règles peuvent être regroupées en deux catégories : celles utilisant les règles inductives et celles utilisant les règles d'association. Ces deux types de règles diffèrent par leur construction.

Définition 2 (Règle) *Une règle est une expression de la forme $\pi : I \rightarrow J$ où $I \subseteq \mathcal{I}$ et $J \subseteq \mathcal{I} \setminus I$. I est appelé antécédent ou corps de la règle et J conséquent. Lorsque J est un attribut classe, π est appelée règle de classe. Un objet $t \in \mathcal{T}$ est couvert par une règle $\pi : I \rightarrow J$ si $\forall i \in I$ on a $\mathcal{R}(t, i) = 1$. L'ensemble des objets couverts par π dans r est noté $couv(\pi, r)$.*

Intuitivement, une règle de classe $\pi : I \rightarrow c$ peut-être interprétée de la manière suivante : si un objet t est décrit par les attributs de I alors t est aussi de classe c. Typiquement, les règles de classe servent à décider la classe de nouveaux objets entrants – de ce fait elles sont aussi appelées règles de décision dans la littérature.

1.2.1 Règles inductives

Dans la littérature, les différentes approches d'apprentissage de règles inductives [QCJ93, Coh95, YH03] suivent l'approche générique par couverture séquentielle décrite dans l'algorithme 1. Par la suite, nous décrivons le fonctionnement des différentes approches et proposons une discussion comparative des classifieurs traités.

Approche générique par couverture séquentielle : L'algorithme de couverture séquentielle est un algorithme de type glouton. A chaque étape, une règle est apprise en utilisant une heuristique (ligne 5), puis les objets couverts par cette nouvelle règle induite sont enlevés de la base (ligne 6), enfin l'algorithme s'arrête lorsque la condition d'arrêt (prenant en compte la couverture totale de la base par les règles) est remplie (ligne 4). Les points clés de cet algorithme sont bien sûr la méthode d'apprentissage des règles, la façon dont les objets couverts par les règles sont enlevés et la condition d'arrêt – et c'est ce que différencie les méthodes existantes. Dans la suite, nous décrivons trois approches d'apprentissage par règles inductives en fonction de ces trois points clés.

Algorithme 1 : Algorithme de couverture séquentielle

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, \ldots c_p\}$ l'ensemble des classes par ordre croissant de taille
Sortie : Π un ensemble de règles induites

1 **begin**
2     $\Pi \leftarrow \emptyset$;
3   **forall** $c_i \in \mathcal{C}$ **do**
4   **while** $\neg$ *ConditionArret* **do**
5             $\pi \leftarrow$ `ApprendreRegle(`$\mathcal{T}, \mathcal{I}, c_i$`)`;
6   Enlever de $\mathcal{T}$ les transactions couvertes par π;
7             $\Pi \leftarrow \Pi \cup \pi$
8     $\Pi \leftarrow \Pi \cup (\pi_d : \emptyset \rightarrow c_k)$
9 **end**

La méthode `FOIL` : First Order Inductive Learner. Introduite dans [QCJ93], `FOIL` est dédiée à la logique du premier ordre. Nous reportons ici la version propositionnelle de `FOIL` adaptée aux contextes binaires.

`FOIL` : Apprentissage de règle. `FOIL` construit ses règles selon l'algorithme 2. Pour la classe courante c_i, P' est l'ensemble courant des objets positifs, i.e. de classe c_i et N' l'ensemble courant des objets négatifs, i.e. des autres classes. Pour construire une règle, `FOIL` part de la règle vide $\pi : \emptyset \rightarrow c_i$ (ligne 2), rajoute successivement le meilleur attribut (selon une mesure d'intérêt) au corps de π (ligne 6) et retire de P' et N' les objets non concernés par la règle en construction, jusqu'à ce qu'il n'y ait plus d'objets négatifs dans N ou que les attributs soient épuisés (ligne 5). La mesure d'intérêt utilisée est la fonction de gain. Pour un attribut a et une règle π est défini comme suit :

$$gain(a, \pi) = |P^*| \cdot \left(\log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|} \right)$$

où $|P|$ (resp. $|N|$) est le nombre d'objets positifs (resp. négatifs) couverts par π et $|P^*|$ (resp. $|N^*|$) le nombre d'objets positifs (resp. négatifs) couverts par la règle π dont le corps a été augmenté de l'attribut a.

Algorithme 2 : `FOIL`-ApprendreRegle

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, \ldots c_p\}$ l'ensemble des classes par ordre croissant de taille
$c_i \in \mathcal{C}$ la classe courante
P l'ensemble des objets positifs (de classe c_i)
N l'ensemble des objets négatifs
Sortie : π une règle induite

1 **begin**
2 $\quad I \leftarrow \emptyset$;
3 $\quad N' \leftarrow N$;
4 $\quad P' \leftarrow P$;
5 $\quad$ **while** $|N'| > 0 \wedge \pi.taille < taille_max_regle$ **do**
6 $\quad\quad$ Trouver l'attribut a qui apporte le plus de gain à π selon P' et N';
7 $\quad\quad I \leftarrow I \cup \{a\}$;
8 $\quad\quad$ Enlever de P' les objets non couverts par π;
9 $\quad\quad$ Enlever de N' les objets non couverts par π;
10 $\quad \pi : I \rightarrow c_i$
11 **end**

`FOIL` : Suppression des transactions couvertes. Après avoir généré une règle π, `FOIL` enlève de r tous les objets de classe c_i couverts par π – donc seulement les objets

positifs.

`FOIL` **: Condition d'arrêt.** `FOIL` s'arrête lorsque tous les objets de classe c_i sont couverts. Il est appliqué pour chacune des classes de r.

La méthode `RIPPER` **:** Repeated Incremental Pruning to Produce Error Reduction. Introduite dans [Coh95], `RIPPER` est une amélioration de l'approche `IREP` (Incremental Reduced Error Pruning) [FW94].

`RIPPER` **: Apprentissage de règle.** `RIPPER` construit ses règles selon l'algorithme 3. Tout d'abord, pour une classe c_i donnée, on différencie l'ensemble P des objets positifs (de classe c_i) de l'ensemble N des objets négatifs. Les objets de la base sont ensuite répartis aléatoirement en respectant la taille des classes en deux sous-ensembles $P_{app} \cup N_{app}$ et $P_{test} \cup N_{test}$ utilisés pour l'accroissement et l'élagage de règles respectivement. Noter que $P_{app} \cup N_{app}$ représente 2/3 de la base courante. Après accroissement à la `FOIL` d'une règle $\pi : I \rightarrow c_i$ en tenant compte de $P_{app} \cup N_{app}$ (ligne 2), celle-ci est immédiatement élaguée en utilisant $P_{test} \cup N_{test}$ de la manière suivante (ligne 3). On considère la mesure suivante pour une règle π construite : $v(\pi, P_{test}, N_{test}) = (p_{test} - n_{test})/(p_{test} + n_{test})$ où p_{test} (resp. n_{test}) est le nombre d'objets de P_{test} (reps. N_{test}) couverts par π. Noter que cette mesure évolue de la même manière que la précision de π sur l'ensemble d'élagage. Puis en partant du dernier attribut a ajouté à π, si $v(\pi' : I \setminus \{a\} \rightarrow c_i, P_{test}, N_{test}) \geq v(\pi : I \rightarrow c_i, P_{test}, N_{test})$ alors on élimine a. Et ainsi de suite pour les autres attributs. Noter que la fonction d'apprentissage de `RIPPER` contient une fonction d'arrêt (ligne 4) qui stoppe l'apprentissage dès lors que le taux d'erreur de la règle en construction est supérieur à 50%.

Algorithme 3 : `RIPPER`-ApprendreRegle

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, \ldots c_p\}$ l'ensemble des classes par ordre croissant de taille
$c_i \in \mathcal{C}$ la classe courante
P l'ensemble des objets positifs (de classe c_i)
N l'ensemble des objets négatifs

Sortie : π une règle induite

```
1 begin
2     π ← FOIL-ApprendreRegle (r, c_i, P_app, N_app);
3     π ← Elaguer (π, P_test, N_test);
4     if Taux_Erreur(π, P_test, N_test) ≥ 50% then
5         return CurrentRuleSet
6 end
```

RIPPER **: Suppression des transactions couvertes.** Lorsqu'une règle π est rajoutée, tous les exemples (positifs comme négatifs) couverts par π sont enlevés de la base.

RIPPER **: Condition d'arrêt.** RIPPER dispose de deux conditions d'arrêt. Première condition : après chaque construction de règle π, si le taux d'erreur de π excède 50% dans $P_{test} \cup N_{test}$, alors π n'est pas rajouté à l'ensemble de règles et RIPPER s'arrête là. L'ensemble construit pour c_i jusqu'alors est l'ensemble de règles finales pour c_i. Deuxième condition : si tous les objets positifs sont couverts, alors RIPPER s'arrête. Dans les deux cas, RIPPER est appliqué aux classes restantes.

Noter que RIPPER dispose aussi de techniques d'optimisation supplémentaires basées sur la longueur minimale de description (MDL : Minimum Description Length) pour décider si certaines règles de l'ensemble final peuvent être remplacées par d'autres règles. Ceci sort de notre cadre de travail. Toutefois les intéressés peuvent se référer à l'article original [FW94].

La méthode CPAR **:** Classification based on Predictive Association Rules. Introduit dans [YH03], CPAR propose deux améliorations par rapport à FOIL et à RIPPER. *(i)* CPAR propose d'apprendre plusieurs règles en même temps. *(ii)* Au lieu d'enlever les objets couverts par une règle induite, les objets couverts sont pondérés de telle sorte qu'ils puissent être couverts à nouveau par de nouvelles règles induites.

CPAR **: Apprentissage de règle.** CPAR construit ses règles selon l'algorithme 4 en utilisant la même fonction de gain que FOIL . Lors de l'accroissement du corps de la règle, seuls les attributs qui apportent un gain supérieur à un gain minimum donné ($gain_minimum = 0.7$) sont retenus (ligne 6). Lorsque plusieurs attributs apportent à peu près le même gain (au plus 1% de différence) à la règle courante (ligne 12), alors plusieurs règles sont générées avec les différents attributs et le processus d'accroissement de chacune des règles continue.

CPAR **: Suppression des transactions couvertes.** A l'initialisation de CPAR, tous les objets positifs (de classe c_i) sont initialisés avec un poids de 1. Ainsi, on a $PoidsDepart(P) = |P|$ qui est aussi le poids total $PoidsTotal(P)$ des objets positifs. Après chaque génération de règle π, on décroît le poids de chaque objet couvert par π en multipliant le poids par un facteur $\alpha = 2/3$ et le $PoidsTotal(P)$ se retrouve diminué. Ainsi chaque objet positif de P pourra être couvert par plusieurs règles induites.

CPAR **: Condition d'arrêt.** Pour $\delta = 0.05$ donné, lorsque $PoidsTotal(P) \leq \delta \times PoidsDepart(P)$ CPAR s'arrête. Notons que les paramètres α et δ sont liés, indiquent le

Algorithme 4 : `CPAR`-ApprendreRegle

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, \ldots c_p\}$ l'ensemble des classes par ordre croissant de taille
$c_i \in \mathcal{C}$ la classe courante
P l'ensemble des objets positifs (de classe c_i)
N l'ensemble des objets négatifs
$I \subseteq \mathcal{I}$ l'ensemble d'attributs de départ de la règle à construire
Sortie : π une règle induite

begin
    $N' \leftarrow N$;
    $P' \leftarrow P$;
    while *true* **do**
        Trouver l'attribut a qui apporte le plus de gain à π selon P' et N';
        if $gain(a, \pi) < gain_minimum$ **then**
            Break;
        else
            $I \leftarrow I \cup \{a\}$;
            Enlever de P' les objets non couverts par π;
            Enlever de N' les objets non couverts par π;
            forall $b \in \mathcal{I} \mid gain(b, \pi)/gain(a, \pi) \geq 99\%$ **do**
                $I \leftarrow I \cup \{b\}$;
                Enlever de P' les objets non couverts par π;
                Enlever de N' les objets non couverts par π;
                `CPAR-ApprendreRegle` (r, c_i, P', N', I);
end

nombre de fois maximum qu'un objet positif peut être couvert en fonction de $|P|$. Les valeurs de ces paramètres sont données par les auteurs.

Discussion : Pour les problèmes à deux classes (c_1, c_2 telles que $|r_{c_1}| \geq |r_{c_2}|$), les algorithmes de génération de règles de `FOIL` et `RIPPER` permettent de générer des règles inductives pour une classe donnée c_1 et une règle par défaut pour la classe majoritaire $\pi : \emptyset \rightarrow c_2$. Pour les problèmes à p classes ($p > 2$), les classes sont ordonnées par ordre croissant de taille. `FOIL` et `RIPPER` est utilisé pour générer un ensemble de règles inductives pour séparer la classe minoritaire c_1 des autres classes $c_2, \ldots, c_p$. Puis les objets couverts par l'ensemble de règles est retiré de r et `FOIL` et `RIPPER` sont utilisés pour générer un autre ensemble de règles inductives pour séparer c_2 des autres classes $c_3, \ldots, c_p$. Le nouvel ensemble de règles est mis à la suite de l'ensemble courant. Et ainsi de suite jusqu'à atteindre la dernière classe majoritaire c_p qui est la classe par défaut – la règle $\pi_{defaut} : \emptyset \rightarrow c_p$ est créée. Notons que cette méthode n'est pas tout à fait de type `OVA` bien que les motifs extraits sont caractéristiques d'une classe par rapport à l'union de plusieurs autres classes. Pour prédire la classe d'un nouvel objet entrant t, l'ensemble de règles est utilisé comme une liste de décision ordonnée par construction, i.e. la première règle supportée par t indique la classe à prédire.
Les faiblesses de `FOIL` et `RIPPER` sont dues au fait que les exemples d'apprentissage ne sont couverts qu'une seule fois, ce qui résulte en un petit ensemble de règles inductives. En raison de la nature même de la procédure `Apprendre-Regle` qui sélectionne successivement le meilleur attribut pour accroître une règle, certaines règles importantes peuvent être oubliées. En effet, la sélection du meilleur attribut occulte d'autres attributs qui peuvent être intéressants (mais un peu moins). De même, la nature de l'algorithme de couverture séquentielle ne garantit pas que l'ensemble final de règles est le meilleur. Le fait de retirer les objets couverts implique que les valeurs de gain calculées par la suite ne sont plus globalement optimales.

`CPAR` au contraire, *(i)* génère des règles inductives pour chacune des classes (en la séparant des autres classes), *(ii)* permet de générer plusieurs règles à la fois si plusieurs attributs apportent un gain similaire à celui du meilleur attribut, *(iii)* permet par un système de pondération de couvrir certains objets avec plusieurs règles. De plus, *(iv)* après génération, chaque règle est évaluée par une estimation de la précision attendue en utilisant l'estimation de l'erreur attendue de Laplace [CB91] :

$$Laplace_estimateur(\pi : I \rightarrow c_i, r) = \frac{n_{c_i} + 1}{n_{total} + p}$$

où n_{total} est le nombre d'objets de r couverts par π, n_{c_i} le nombre d'objets de r_{c_i} couverts par π et p le nombre classes. Puis, *(v)* pour prédire la classe d'un nouvel objet t entrant, `CPAR` sélectionne les k meilleures règles selon l'estimateur de Laplace qui couvrent t pour chaque classe. La classe qui maximise la valeur moyenne de l'estimateur indique la classe à prédire.

Ainsi, parmi les différentes approches par règles inductives, CPAR est la méthode la plus récente, semble la plus évoluée et la plus performante au vu des résultats de précision annoncés dans l'article original. Toutefois, bien que CPAR génère plus de règles que ces concurrents, le système de pondération n'assure pas d'avoir les meilleures règles pour chaque objet. De plus, CPAR dépend d'un paramétrage plus lourd. En effet, le *gain_minimum*, le facteur de pondération α, la condition d'arrêt paramétrée par δ, et le nombre de règles à utiliser k sont loin d'être intuitifs pour l'utilisateur et dépendra du domaine de travail.

1.2.2 Classification associative

Règles d'association, itemsets fréquents et extraction. La classification associative est une méthode de classification supervisée basée sur les règles d'association. Avant de discuter de ces méthodes nous rappelons brièvement les travaux pionniers sur l'extraction des itemsets fréquents et des règles d'association. Lors de leur introduction dans [AIS93, AS94], les auteurs proposent d'extraire les règles d'associations valides en utilisant l'ensemble des itemsets fréquents (voir définitions 3 et 4).

Définition 3 (Itemset, Itemset fréquent, Support) *Un itemset $I \subseteq \mathcal{I}$ est un sous-ensemble d'attributs de $\mathcal{I}$. La fréquence d'un itemset $I \subseteq \mathcal{I}$ est $freq(I,r) = |Objets(I,r)|$, où $Objets(I,r) = \{t \in \mathcal{T} | \forall i \in \mathcal{I} : \mathcal{R}(t,i) = 1\}$ est appelé support de I et noté $supp(I,r)$. Étant donné un entier positif γ, un itemset est dit γ-fréquent si $freq(I,r) \geq \gamma$. Par la suite, nous utiliserons aussi la notion de fréquence relative d'un itemset I qui est $freq_r(I,r) = freq(I,r)/|r|$.*

Définition 4 (Règle d'association) *Une règle d'association dans r est une expression de la forme $\pi : I \rightarrow J$ où $I \subseteq \mathcal{I}$ et $J \subseteq \mathcal{I} \setminus I$. La fréquence d'une telle règle π dans r est $freq(\pi,r) = freq(I \cup J, r)$ et sa confiance $conf(\pi,r) = freq(I \cup J,r)/freq(I,r)$. Soit min_freq et min_conf deux valeurs de seuil données, la règle d'association π est dite valide si $freq(\pi,r) \geq min_freq$ et $conf(\pi,r) \geq min_conf$. Lorsque J est un attribut classe c, $\pi : I \rightarrow c$ est appelée règle d'association de classe.*

En effet, soit $I \subseteq \mathcal{I}$ un itemset tel que $freq(I,r) \geq min_freq$. I peut être divisé en deux parties, un conséquent Y et un corps de règle $X = I \setminus Y$ pour former la règle fréquente $\pi : X \rightarrow Y$. Le processus de découverte de règles d'association valides à partir de I est itératif. Tout d'abord on considère le cas $Y = \emptyset$. Dans ce cas, $I \rightarrow \emptyset$ est valide car fréquente et de confiance maximale 1. Puis, est généré l'ensemble des candidats conséquents C_{k+1} de taille $k+1$ en partant de $k = 0$. On sait qu'un conséquent est candidat si tous ses ensembles sont conséquents de règles confiantes (et donc valides). Pour calculer la confiance d'une règle candidate, on peut utiliser les fréquences de I et de X calculées lors de l'extraction des itemsets fréquents.

Depuis, les efforts de recherche se sont focalisés sur les algorithmes d'extraction d'itemsets fréquents qui semble être le point critique pour la tâche d'extraction de règles d'association. Dans [AS94], les auteurs proposent l'algorithme de recherche en largeur `APRIORI` décrit dans l'algorithme 5.

Algorithme 5 : `APRIORI`

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
min_freq un seuil de fréquence minimum
Sortie : $\mathbb{F}$ l'ensemble des itemsets fréquents de r

begin
  $C_1 = \{\{i\}|i \in \mathcal{I}\}$;
  $k = 1$;
  while $C_k \neq \emptyset$ **do**
    Calculer la fréquence des itemsets candidats de C_k;
    $\mathbb{F}_k = \{X \in C_k | freq(X, r) \geq min_freq\}$;
    Générer l'ensemble C_{k+1} des itemsets candidats de taille $k+1$;
    $k = k + 1$;
  $\mathbb{F} = \bigcup_{j=1}^{j=k} \mathbb{F}_j$;
end

`APRIORI` effectue une recherche en largeur en générant les itemsets candidats C_{k+1} de taille $k+1$ en commençant avec $k = 0$. un itemset est candidat si tous ses sous-ensembles sont fréquents. Au départ, C_1 contient tous les attributs de I, puis pour un niveau k, les candidats sont générés de la manière suivante : *(i)* pour $X, Y \in \mathbb{F}_k$, on considère l'union $X \cup Y$ si X et Y contiennent un $(k-1)$-itemset en commun. *(ii)* Puis $X \cup Y$ est inséré dans C_{k+1} si tous leurs sous-ensembles directs appartiennent à $\mathbb{F}_k$. Le calcul des fréquences des itemsets candidats se fait en une seule passe sur l'ensemble des objets $\mathcal{T}$ de r. A chaque objet $t \in \mathcal{T}$, la fréquence de chaque itemset candidat couvrant l'objet est mis à jour. Puis tous les k-itemsets fréquents sont insérés dans $\mathbb{F}_k$. Noter que la génération des candidats avec `APRIORI` profite de la propriété d'anti-monotonicité de la fréquence. C'est-à-dire, si un itemset I n'est pas fréquent, aucun de ses sur-ensembles ne seront fréquents ; ce qui permet d'élaguer l'espace de recherche. Cette approche rentre parfaitement dans le cadre des algorithmes par niveaux et de la théorie des bordures introduit dans [MT97] que nous détaillerons dans le chapitre 2.

Puisque nous nous intéressons aux règles d'association de classe que nous dérivons à partir des itemsets fréquents, nous pouvons nous intéressés uniquement aux itemsets fréquents contenant un attribut classe. Pour ce faire, il suffit d'initialiser l'ensemble des candidats C_1 à $\mathcal{C}$ l'ensemble des classes. Ainsi les itemsets fréquents générés par la suite contiendront un attribut classe. Ensuite, pour la génération des règles d'association de

classe, les candidats conséquents de règles sont restreints aux attributs classe. Si deux règles ont le même corps mais concluent sur deux classes différentes, le conflit est résolu par les valeurs de confiance ; la règle la plus confiante est gardée.

Dans la suite nous décrivons deux approches pionnières de classification supervisée basées sur les règles d'association valides : CBA et CMAR [LHM98, LHP01]. Toutes deux procèdent selon le même schéma : tout d'abord l'ensemble des règles d'association valides est extrait, puis les règles redondantes et les moins intéressantes sont retirées de cet ensemble. Enfin, à partir de cet ensemble post-traité, un classifieur est construit.

La méthode CBA : Classification Based on Associations. Introduite dans [LHM98], CBA utilise un algorithme de type APRIORI pour extraire les règles d'association de classe valides en fonction de deux seuils de fréquence (min_freq) et de confiance (min_conf). A chaque règle extraite $\pi : X \rightarrow c_i$, CBA teste si le taux d'erreur pessimiste [Qui93] de π est supérieur au taux d'erreur pessimiste d'une règle π' dont le corps est un sous-ensemble direct de X. Si c'est le cas, π n'est pas gardé dans l'ensemble final de règles.

CBA : Élagage de l'ensemble de règles. Pour réduire l'ensemble de règles Π, CBA classe d'abord les règles selon l'ordre suivant : soient $\pi_1 : X \rightarrow c_i$ et $\pi_2 : Y \rightarrow c_j$ deux règles de Π, $\pi_1 \succ \pi_2$ si *(i)* $conf(\pi_1, r) > conf(\pi_2, r)$, *(ii)* si $conf(\pi_1, r) = conf(\pi_2, r)$ mais $freq(\pi_1, r) > freq(\pi_2, r)$, *(iii)* si à confiance et fréquence égale mais $|X| < |Y|$. Puis, l'ensemble de règles est élagué en fonction de leur ordre dans Π et de leur couverture des objets de la base afin d'éliminer les redondances entre règles et ne garder que les meilleures règles selon $\succ$. Ainsi, chaque objet t de la base est couvert par la meilleure des règles qui couvrent t (selon $\succ$). De plus, lorsqu'une règle est choisie, c'est qu'elle classifie correctement au moins un objet de la base. Enfin, l'ensemble final Π_C des règles constitue le classifieur et pour prédire la classe d'un nouvel objet entrant t, la première règle qui couvre t indique la classe à prédire.

La méthode CMAR : (Classification based on Multiple Association Rules). Introduite dans [LHP01], CMAR s'appuie aussi sur l'ensemble des règles d'association de classe valides pour construire un classifieur. Pour extraire les règles d'association valides CMAR utilise l'algorithme FP-Growth [HPY00]. Bien que différent de l'approche APRIORI, FP-Growth génère le même ensemble de règles puisque que la tâche d'extraction de règles valides est un problème déterministe.

CMAR : Élagage des règles. CMAR considère le même ordre que CBA sur les règles d'associations de classe valides. De plus on considère qu'une règle $\pi_1 : X \rightarrow c_i$ est plus générale que $\pi_2 : Y \rightarrow c_j$ si $X \subseteq Y$. CMAR propose trois types d'élagage : *(i)* on préfère

les règles générales de forte confiance plutôt que les règles spécifiques à faible confiance. Ainsi π_2 sera élagué si $\pi_1 \succ \pi_2$ et π_1 est plus générale que π_2. *(ii)* `CMAR` sélectionne uniquement les règles positivement corrélées en fonction du test du χ^2. *(iii)* `CMAR` élague encore l'ensemble de règles en fonction de leur couverture des objets de la base afin que certains objets soient couverts au moins par $\delta = 4$ règles (δ est donné par les auteurs). Enfin, pour classer un nouvel objet entrant t, `CMAR` regroupe les règles supportées par t selon leur conséquent (la classe), puis mesure l'effet combiné de chaque groupe en calculant la valeur du χ^2 pondéré ; et le groupe ayant la plus grande valeur d'effet combiné indique la classe.

Discussion : Les approches de classification supervisée basée sur les règles d'associations de classe valides sont confrontées à l'explosion du nombre de motifs extraits. De plus, l'ensemble extrait contient des motifs peu intéressants, et des motifs redondants (i.e. couvrant les mêmes objets ou avec le même pouvoir discriminant tout en étant en relation via $\subseteq$) qui oblige une étape de post-traitement pour éliminer les motifs indésirables. De plus, la confiance qui est la mesure d'intérêt phare de ces méthodes ne tient pas compte de la distribution des classes. Ainsi, pour un problème à deux classes (c_i, c_j), deux règles valides $\pi : X \rightarrow c_i$ et $\pi' : Y \rightarrow c_j$ de même fréquence et confiance caractérisant deux classes différentes à distributions inégales ($|c_i| \gg |c_j|$) auront le même pouvoir discriminant alors que leur taux d'erreurs relatives seront très différents : $freq_r(X, r_{c_j}) > freq_r(Y, r_{c_i})$ (où r_c est la base de données restreinte aux objets de classe c). Dans la suite, nous discutons des travaux de la littérature basées sur les itemsets émergents qui tiennent compte de la distribution des classes et traite ce problème.

1.3 Méthodes à base d'itemsets émergents

Dans [DL99], les auteurs introduisent un nouveau type de motif local, les itemsets émergents. Intuitivement, un itemset émergent pour une classe c_i apparaît plus fréquemment dans les données de la classe c_i que dans le reste de la base (on parle ici de fréquence relative). Plus formellement, un itemset émergent pour une classe c_i est défini comme suit :

Définition 5 (Taux d'accroissement et itemset ρ-émergent) *Soit $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ une base de données binaires, et $\mathcal{C} = \{c_1, \ldots, c_p\}$ l'ensemble des attributs classe. Le taux d'accroissement d'un itemset $I \subseteq \mathcal{I}$ pour la classe c_i dans r est :*

$$TA(I, r_{c_i}) = \begin{cases} 0 & si\ freq_r(I, r_{c_i}) = 0 \\ \infty & si\ freq_r(I, r_{c_i}) > 0 \quad \wedge freq_r(I, r_{c_j}) = 0 \quad (\forall j \neq i) \\ \frac{freq_r(I, r_{c_i})}{freq_r(I, r \setminus r_{c_i})} & sinon \end{cases}$$

où r_{c_i} est la base de données restreinte aux objets de classe c_i. Pour un entier $\rho > 1$, on dit que I est un itemset ρ-émergent (ρ-EP)[1] *pour la classe c_i dans r si $TA(I, r_{c_i}) \geq \rho$. Lorsque $TA(I, r_{c_i}) = \infty$, i.e. I n'est supporté que par des objets de la classe c_i, on dit que I est un Jumping Emergent Pattern (JEP). Soit un entier $\gamma > 0$, on parle de ρ-EPs ou de JEPs γ-fréquents pour des motifs émergents par rapport à un seuil de fréquence γ.*

L'extraction de l'ensemble des itemsets ρ-émergents est un problème difficile. En effet, il s'agit d'extraire les itemsets qui sont fréquents dans une classe et infréquents dans le reste des données. Ici, la propriété d'anti-monotonicité ne tient plus car si un itemset I n'est pas émergent, il se peut que $I \cup \{i\}$ soit émergent. Dans [DL99], les auteurs proposent un algorithme d'extraction des itemsets émergents pour une classe c_i en utilisant deux bordures : *(i)* l'ensemble des itemsets infréquents minimaux (au sens de l'inclusion) et *(ii)* l'ensemble des itemsets fréquents maximaux. Ainsi, les itemsets dans l'intervalle de ces deux bordures sont des itemsets émergents. Bien que les itemsets émergents soient accessibles via ces deux bordures, pour calculer le taux d'accroissement d'un itemset émergent, le retour aux données est inévitable ; en effet, si on sait qu'un itemset est émergent parce qu'il est fréquent dans une classe et infréquent dans le reste de la base, les valeurs de fréquence sont inconnues et donc le taux d'accroissement est inconnu aussi. De plus l'algorithme d'extraction des itemsets ρ-émergents doit être appliqué pour chacune des classes de la base. Dans la suite, nous exposons deux méthodes de classification basées sur la notion d'émergence, `CAEP` [DZWL99] utilisant les ρ-EPs et `JEPC` [LDR00b] utilisant les JEPs.

La méthode `CAEP` : Classification by Aggregating Emerging Patterns. Introduite dans [DZWL99], `CAEP` fonctionne de la manière suivante. *(i)* Étant donnés deux seuils de fréquence γ et de taux d'accroissement ρ, pour chaque classe c_i, `CAEP` extrait l'ensemble S_{c_i} des itemsets γ-fréquents ρ-émergents pour c_i. *(ii)* Comme S_{c_i} peut être très grand, contenir des redondances et des EPs moins intéressants, `CAEP` ne garde que les EPs dits essentiels dans S'_{c_i}. Tout d'abord, S_{c_i} est ordonné selon le taux d'accroissement (ρ) puis la fréquence (γ). Puis, l'ensemble S'_{c_i} est initialisé avec le premier élément de S_{c_i}. Pour chaque autre itemset $s \in S_{c_i}$, on remplace tous les éléments $s' \in S_{c_i}$ par s si $s \subset s'$ et

- *(a)* $TA(s, r_{c_i}) \geq TA(s', r_{c_i})$
- *(b)* ou $freq(s, r) \gg freq(s', r)$ et $TA(s, r_{c_i}) \geq \rho'$ pour un $\rho' > \rho$ donné.

Si *(a)* et *(b)* ne sont pas vérifiés et s n'est le sur-ensemble d'aucun itemset s' de S'_{c_i}, alors on ajoute s dans S'_{c_i}. Enfin, pour un nouvel objet t, `CAEP` prend en compte tous les EPs qui couvrent t et calcule le score suivant pour chaque classe c_i :

$$scoreEP(t, c_i) = \sum_{\{s \in S'_{c_i} | s \subseteq t\}} \frac{TA(s, r)}{TA(s, r) + 1} \times freq_r(s, r_{c_i})$$

1. EP est le sigle de Emerging Pattern pour motif émergent en anglais.

Enfin, après normalisation des scores, la classe obtenant le plus haut score est la classe à prédire pour t.

La méthode JEPC : Jumping Emerging Patterns based Classifier. Introduit dans [LDR00b, LDR01], JEPC se focalise sur les JEPs, ces itemsets qui n'apparaissent que dans une seule classe des données. L'extraction des bordures pour les JEPs se fait de la même manière qu'avec CAEP. Toutefois, ici, la post-sélection des JEPs ne se fait plus en fonction de TA (puisque les JEPs sont équivalents par rapport à TA) mais en fonction de leurs fréquences. Ainsi, les JEPs à forte fréquence sont préférées et forment l'ensemble des JEPs les plus expressifs pour une classe c_i ($\texttt{MEJEP}(r_{c_i})$ [2]) ; ceux-ci se trouvent sur les bordures des infréquents minimaux construites lors de l'extraction. Enfin, pour un nouvel objet t, JEPC calcule pour chaque classe c_i un score représentant l'impact collectif des JEPs qui couvrent t selon la formule suivante :

$$ImpactCollectif(t, c_i) = \sum_{I \in \texttt{MEJEP}(r_{c_i}) \wedge I \subseteq t} freq_r(I, r_{c_i})$$

Le plus haut score obtenu indique la classe à prédire pour t.

Discussion : Il existe d'autres méthodes de classification à base d'EPs [FR03, LDRW04]. D'autre part, dans [RF07], il est proposé un résumé des différentes approches à base d'EPs. Contrairement aux approches basées sur les règles d'association valides, les méthodes à base d'EPs, par définition du taux d'accroissement, tiennent compte d'une certaine manière de la répartition des classes. Ainsi, le cadre des EPs semble plus approprié que le cadre fréquence-confiance pour certaines tâches de classification difficiles où les classes sont disproportionnées. Toutefois, l'extraction de l'ensemble des EPs est une tâche difficile. Les premiers algorithmes se limitent aux γ-fréquents EPs [DL99] ou aux JEPs [LRD00]. Plus tard, dans [ZDR00, BMR02], d'autres algorithmes plus efficaces sont développés pour améliorer l'extraction des EPs. Malgré tout, l'ensemble des EPs peut contenir des éléments moins intéressants que d'autres et redondants. C'est pourquoi, un post-traitement est nécessaire. Dans la phase de post-traitement, CAEP préfère les EPs généraux I à fort taux d'accroissement ou très fréquents plutôt que leurs sur-ensembles $J \supseteq I$. JEPC quant à lui, préfère les JEPs généraux les plus fréquents. Ainsi, le problème de la redondance est pris en compte en post-traitement.

1.4 Limites

Nous avons identifié les faiblesses de certaines approches de classification supervisée à base de motifs locaux. Les approches par règles inductives n'assurent pas d'obtenir l'en-

2. MEJEP : Most Expressive Jumping Emerging Patterns pour l'ensemble des JEPs les plus expressifs.

semble des meilleures règles en raison de la nature même de l'algorithme de couverture séquentielle. Les approches par règles d'association au contraire misent sur un ensemble exhaustif (ou presque) des règles valides. L'extraction est alors coûteuse, l'ensemble résultant est vaste et contient des motifs indésirables car moins intéressants et redondants par rapport à d'autres. Dans le chapitre 2, nous replaçons l'extraction de motifs fréquents dans le cadre de travail de [MT97], nous discutons des différentes représentations condensées des motifs fréquents, que nous utilisons pour traiter la redondance en classification supervisée dans le chapitre 3.

Dans [WK05, WK06], pour éviter la phase exhaustive d'extraction et de post-traitement, les auteurs proposent un nouveau cadre de travail centré sur les objets de la base. Ils extraient directement les k règles les plus confiantes pour chaque objet. Toutefois, la confiance, ne prenant pas en compte la distribution des classes, n'est pas très appropriée pour les problèmes aux classes disproportionnées. Les approches à base d'EPs apportent une solution à ce problème avec le taux d'accroissement des itemsets.

D'autre part, les approches basées sur les règles d'association valides comme les approches basées sur les EPs sont des approches de type `OVA` dans le sens où les motifs extraits caractérisent une classe par rapport au reste de la base. Bien que les méthodes à base d'EPs prennent en compte d'une certaine manière la distribution des classes, elles ne tiennent pas compte de la répartition des erreurs des EPs dans les différentes classes du reste de la base. Ainsi, un EP pour une classe c_i peut être corrélé positivement avec une classe $c_j (j \neq i)$, ce qui peut générer des incohérences et des conflits entre motifs dans les problèmes multi-classes à distributions inégales. Dans le chapitre 4, nous nous attaquons à ce problème et proposons une approche `OVE` où la répartition des erreurs dans les différentes classes est prise en compte.

Chapitre 2

Représentations condensées des itemsets fréquents

Sommaire

2.1 Théories, bordures et représentations condensées

L'extraction des itemsets fréquents est une tâche essentielle de la fouille de données. L'ensemble des itemsets fréquents résument en quelque sorte les tendances sous-jacentes aux données. Ainsi, les itemsets fréquents nous permettent de trouver des motifs intéressants "cachés" dans les données comme par exemple des règles d'association, des séquences, des épisodes, des regroupements, ou encore des classifieurs. Dans ce chapitre,

nous replaçons la tâche d'extraction d'itemsets fréquents dans le cadre de travail de Manilla et Toivonen [MT97]. Dans un tel cadre, calculer l'ensemble des itemsets fréquents revient à calculer la théorie $Th(\mathcal{D}, \mathcal{L}, \mathbb{C}) = \{\phi \in \mathcal{L} \mid \mathbb{C}(\phi, \mathcal{D})\}$ où $\mathcal{D}$ est une base de données et correspond ici à notre contexte binaire $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$, $\mathcal{L}$ est un langage de motifs et correspond ici à $\mathcal{P}(\mathcal{I})$ l'ensemble des parties de $\mathcal{I}$, $\mathbb{C}$ est un prédicat de sélection ou contrainte et est réduit ici à une contrainte de fréquence minimum. Ainsi, $Th(\mathcal{D}, \mathcal{L}, \mathbb{C})$ devient $Th(r, \mathcal{P}(\mathcal{I}), \mathbb{C}_{\gamma-minfreq}) = \{I \in \mathcal{P}(\mathcal{I}) \mid freq(I, r) \geq \gamma\}$.

Après l'introduction du problème d'extraction d'itemsets fréquents (`FIM`[1]) [AIS93], les premières approches furent développées pour extraire de manière efficace tous les itemsets fréquents d'une base de données r. Notons deux familles différentes de techniques de parcours de l'espace de recherche : *(i)* les méthodes de parcours en largeur et *(ii)* les méthodes de parcours en profondeur. Toutes deux profitent de la propriété d'anti-monotonicité de la contrainte de fréquence minimum $\mathbb{C}_{\gamma-minfreq}$ pour éviter des parties de l'espace de recherche. Pour rappel :

Définition 6 (Contrainte anti-monotone) *Soit* $\mathbb{C} : \mathcal{P}(\mathcal{I}) \mapsto \{vrai, faux\}$ *une contrainte.* $\mathbb{C}$ *est une contrainte anti-monotone dans* r *si et seulement si :*

$$\forall (I, J) \in 2^{\mathcal{I}} : \mathbb{C}(I, r) \wedge J \subseteq I \Rightarrow \mathbb{C}(J, r)$$

Après le premier algorithme `AIS` [AIS93], plusieurs autres algorithmes ont vu le jour et ont apporté de multiples améliorations en terme de performance : par exemple, `APRIORI` [AS94] présenté au chapitre précédent permet un meilleur élagage de l'espace de recherche ; ou encore `FP-Growth` évite une génération explicite des itemsets candidats. Il en existe beaucoup d'autres, les meilleurs résultant de deux challenges [GZ03, GZ04, BGZ04] sont détaillés et accessibles sur la page des challenges `FIMI` : http://fimi.cs.helsinki.fi/. Notons que certains de ces derniers algorithmes combinent plusieurs améliorations de différentes approches pour améliorer les performances de l'extraction.

Toutefois, le nombre d'itemsets fréquents est souvent trop grand. Le stockage des FI et le calcul de leur fréquence n'est pas supporté par les calculateurs d'aujourd'hui. C'est le cas lorsque le seuil de fréquence minimum est très bas ou lorsque les données sont très corrélées. En effet, dans le pire des cas, le nombre d'itemsets fréquents dans r est exponentiel par rapport au nombre d'attributs (complexité exponentielle : $O(2^I)$). Il est possible d'éviter cette explosion combinatoire en calculant des représentations condensées des itemsets fréquents. Le principe est de calculer un ensemble $CR \subseteq \mathcal{P}(\mathcal{I})$ le plus concis possible à partir duquel nous pouvons déduire efficacement $Th(r, \mathcal{P}(\mathcal{I}), \mathcal{C}_{\gamma-minfreq})$, i.e. sans accéder à nouveau à r.
Une première solution consiste en le calcul des bordures (positive ou négative) de la théorie $Th(r, \mathcal{P}(\mathcal{I}), \mathcal{C}_{\gamma-minfreq})$. Pour simplifier les notations de notre problème, nous renommons

1. `FIM` pour Frequent Itemset Mining en anglais.

la théorie $Th(r, \mathcal{P}(\mathcal{I}), \mathcal{C}_{\gamma-minfreq})$ en $\mathbb{F}(\gamma, r)$ – i.e. l'ensemble des itemsets γ-fréquents de r. La bordure positive de $\mathbb{F}(\gamma, r)$, notée $\mathcal{B}d^+(\mathbb{F}(\gamma, r))$, est l'ensemble des plus grands itemsets γ-fréquents (au sens de l'inclusion) de $\mathbb{F}(\gamma, r)$ et est donc définie comme suit :

$$\mathcal{B}d^+(\mathbb{F}(\gamma, r)) = \{I \in \mathbb{F}(\gamma, r) \mid \forall J \in \mathcal{P}(\mathcal{I}) : I \subset J \Rightarrow J \notin \mathbb{F}(\gamma, r)\}$$

Notons qu'une méthode d'extraction des itemsets fréquents maximaux est proposée dans [Bay98]. $\mathcal{B}d^+(\mathbb{F}(\gamma, r))$ constitue déjà une première représentation condensée de $\mathbb{F}(\gamma, r)$. En effet, $\mathcal{B}d^+(\mathbb{F}(\gamma, r)) \subset \mathbb{F}(\gamma, r)$ et tous les sous-ensembles des itemsets fréquents maximaux sont fréquents et peuvent être déduits sans retour aux données. Cependant, dans la majorité des applications, en plus des itemsets fréquents, on veut aussi connaître leur fréquence. Par exemple, si l'on désire dériver des règles d'association à partir des itemsets fréquents, la fréquence des itemsets est indispensable pour le calcul de mesure d'intérêt des règles comme la confiance.
La bordure négative de $\mathbb{F}(\gamma, r)$, notée $\mathcal{B}d^-(\mathbb{F}(\gamma, r))$, est l'ensemble des plus petits itemsets qui ne sont pas γ-fréquents et est définie comme suit :

$$\mathcal{B}d^-(\mathbb{F}(\gamma, r)) = \{I \in \mathcal{P}(\mathcal{I}) \setminus \mathbb{F}(\gamma, r) \mid \forall J \in \mathcal{P}(\mathcal{I}) : J \subset I \Rightarrow J \in \mathbb{F}(\gamma, r)\}$$

Bien que tout sous-ensemble strict de $\mathcal{B}d^-(\mathbb{F}(\gamma, r))$ est fréquent, $\mathcal{B}d^-(\mathbb{F}(\gamma, r))$ ne sera pas considérée comme une représentation condensée de $\mathbb{F}(\gamma, r)$ car $\mathcal{B}d^-(\mathbb{F}(\gamma, r)) \not\subseteq \mathbb{F}(\gamma, r)$. Toutefois la définition de la bordure négative nous sera utile pour la présentation des autres représentations condensées.

Ainsi, une représentation condensée CR qui permet de déduire tous les itemsets fréquents mais pas leur valeur de fréquence sera dite non-informative. Par exemple, $\mathcal{B}d^+(\mathbb{F}(\gamma, r))$ est non-informative. Si les valeurs de fréquence peuvent être déduites, on distinguera deux cas : lorsque CR permet de déduire de manière exacte la valeur de fréquence de chaque itemset fréquent, on dit que CR est une *représentation condensée exacte*, si seulement une valeur approximative est accessible alors CR est appelée *représentation condensée approximative*. Outre l'informativité, une représentation condensée pourra aussi être qualifiée par :
- sa taille (le plus petit étant le meilleur),
- l'efficacité et la complétude des algorithmes permettant de la générer,
- ainsi que la rapidité avec laquelle on peut générer des informations intéressantes à partir d'elle (e.g. les itemsets fréquents, les règles d'association intéressantes,. . .).

Dans la suite, nous exposons les différents concepts-clés développés ces dernières années pour le calcul de représentations condensées d'itemsets fréquents. Nous traiterons entre autres des représentations condensées basées sur les itemsets fermés [BB00, PBTL98, PBTL99], les itemsets δ-libres [BBR00, BBR03], les itemsets $\vee$-libres [BR01, BR03] et leur généralisation [KG02], les itemsets non-dérivables [CG02, CG07] ainsi qu'un cadre unificateur [CG03].

2.2 Les itemsets fermés

Les notions d'itemset fermé et de fermeture d'un itemset sont issues de la théorie des treillis [Bir67, BM70], plus précisément de l'analyse de concepts formels [GSW05]. L'utilisation de cette théorie pour l'extraction d'itemsets fréquents dans une base de données binaires r a été initiée par Pasquier et al. [PBTL98, PBTL99]. Formellement, un itemset fermé est défini comme suit :

Définition 7 (Itemset fermé et fermeture) *Un itemset $I \subseteq \mathcal{I}$ est dit fermé (ou clos) dans r si et seulement si il n'existe pas de sur-ensemble J de I ayant le même support (et donc la même fréquence) que I, c'est-à-dire :*

$$\nexists J \subseteq \mathcal{I} : I \subset J \wedge freq(I,r) = freq(J,r)$$

La fermeture d'un itemset $I \subseteq \mathcal{I}$ dans r, notée $cl(I,r)$, est l'unique sur-ensemble maximal (selon $\subseteq$) de I qui a le même support que I. Notons qu'un itemset fermé est égal à sa fermeture dans r.

Ainsi, étant donné un seuil de fréquence minimum $\gamma > 0$, la seule connaissance de l'ensemble des itemsets fréquents fermés et de leur fréquence dans r nous suffit pour générer tous les itemsets fréquents $\mathbb{F}(\gamma, r)$ ainsi que leur support. L'ensemble des itemsets fréquents fermés de r, noté $\mathbb{F}(\gamma, cl, r)$, constitue une représentation condensée de $\mathbb{F}(\gamma, r)$. En effet, soit un itemset $I \subseteq \mathcal{I}$. Si I n'a pas de sur-ensemble dans $\mathbb{F}(\gamma, cl, r)$, alors $cl(I,r)$ n'est pas fréquent et par conséquent, I ne peut être fréquent. Au contraire, s'il existe au moins un sur-ensemble de I dans $\mathbb{F}(\gamma, cl, r)$, alors $freq(I,r) = freq(J,r)$ où $J = cl(I,r)$ – J est en fait le plus petit sur-ensemble de I dans $\mathbb{F}(\gamma, cl, r)$.

Comme exemple, considérons la base de données binaires r décrite par la table 2.1.

r	a	b	c	d	e
t_1	1	1	1	1	1
t_2	0	0	1	1	0
t_3	1	0	1	1	1
t_4	0	1	1	1	1
t_5	1	1	1	1	0
t_6	1	0	1	1	1

TABLE 2.1 – Exemple de base de données binaires r.

Dans cet exemple, l'itemset ace n'est pas fermé car son sur-ensemble direct $acde$ est de même fréquence (3). Par contre, l'itemset $acde$ est l'unique sur-ensemble maximal de ace ayant la même fréquence, c'est pourquoi $cl(ace, r) = acde$. Considérons maintenant un

seuil de fréquence minimum $\gamma = 3$. L'ensemble des itemsets fréquents fermés dans r est $\mathbb{F}(\gamma, cl, r) = \{acd, acde, bcd, bcde, cde\}$. Leur fréquence respective est 4,3,3,6,4. Ces deux seules connaissances nous permettent de générer tout itemset fréquent X et sa valeur de fréquence en considérant le plus petit élément de $\mathbb{F}(\gamma, cl, r)$ qui est un sur-ensemble de X : par exemple, nous savons que ae est fréquent car $cl(ae, r) = acde$ et $acde \in \mathbb{F}(\gamma, cl, r)$; nous savons aussi que $freq(ae, r) = 3$ car $freq(acde, r) = 3$.

2.3 Les itemsets δ-libres

La notion d'itemset δ-libre a été introduite la première fois dans [BBR00, BBR03]. Elle fait appel à la notion de règle d'association δ-forte. Intuitivement, une règle d'association δ-forte est une règle dont le nombre d'erreurs commises dans la base est borné par un entier $\delta > 0$ (généralement petit par rapport à $|\mathcal{T}|$). Plus formellement, règle δ-forte et itemset δ-libre se définissent comme suit :

Définition 8 (règle d'association δ-forte, itemset δ-libre) *Une règle d'association δ-forte est une règle d'association de la forme $\pi : I \rightarrow^{\delta} a$, où $I \subseteq \mathcal{I}$, $a \in \mathcal{I} \setminus I$ et δ un entier naturel. On dit que π est valide dans r si $freq(I, r) - freq(I \cup \{a\}, r) \leq \delta$, en d'autres termes, s'il y a moins de δ transactions où π est violée.*
Un itemset $J \subseteq \mathcal{I}$ est un itemset δ-libre si et seulement si il n'existe pas de règle δ-forte valide $\pi : I \rightarrow^{\delta} a$ telle que $I \subset J$, $a \in J$ et $a \notin I$.

Soit $\mathbb{F}(\gamma, \delta, r)$ l'ensemble des itemsets γ-fréquents δ-libres de r. La connaissance des fréquences des éléments de $\mathbb{F}(\gamma, \delta, r)$ nous permet d'approximer la fréquence des itemsets fréquents de r qui ne sont pas δ-libres. En effet, soit J un itemset fréquent non δ-libre. Par définition, il existe une règle δ-forte $\pi : I \rightarrow^{\delta} a$ telle que $I \subset J$ et $a \in J$. De plus, si π est δ-forte valide, alors $J \setminus \{a\} \rightarrow^{\delta} a$ est aussi valide. De ce fait, on peut approximer la fréquence de J par la fréquence de l'itemset fréquent $J \setminus \{a\}$. En effet, $freq(J \setminus \{a\}, r) - \delta \leq freq(J, r)$ indique que $freq(J \setminus \{a\}, r)$ est une borne supérieure pour $freq(J, r)$. Ainsi, si $J \setminus \{a\}$ est γ-fréquent δ-libre, l'approximation est donnée, sinon on étudiera une approximation par la fréquence d'un itemset de plus petite taille. Lorsque plusieurs bornes supérieures existent, la plus petite sera la plus précise et celle qu'on choisira en pratique sera issue de la plus petite valeur de fréquence des itemsets γ-fréquents δ-libres inclus dans J. Notons que l'erreur commise lors de l'approximation est facteur de δ et est petite en pratique [BBR03]. De plus, lorsque $\delta = 0$, les fréquences déduites sont exactes.
Si $\mathbb{F}(\gamma, \delta, r)$ nous permet d'approximer la fréquence de tous les itemsets γ-fréquents non δ-libres, cela ne nous permet pas de dire si un itemset est γ-fréquent. Pour y remédier, nous utilisons l'ensemble des itemsets non-fréquents δ-libres minimaux en plus de $\mathbb{F}(\gamma, \delta, r)$.

Notons tout d'abord que la propriété de δ-liberté est anti-monotone – en effet, par contraposée, si un itemset X n'est pas δ-libre, alors il existe une règle d'association δ-forte valide entre deux de ses sous-ensembles et qui sera aussi valide dans tout sur-ensemble $Y \supseteq X$. Nous pouvons donc définir la bordure négative de $\mathbb{F}(\gamma, \delta, r)$ comme suit :

$$\mathcal{B}d^-(\mathbb{F}(\gamma, \delta, r)) = \{I \in \mathcal{P}(\mathcal{I}) \setminus \mathbb{F}(\gamma, \delta, r) \mid \forall J \in \mathcal{P}(\mathcal{I}) : J \subset I \Rightarrow J \in \mathbb{F}(\gamma, \delta, r)\}$$

Les éléments de $\mathcal{B}d^-(\mathbb{F}(\gamma, \delta, r))$ sont ou non fréquents ou non δ-libres. Donc, si l'on note $\mathbb{F}(\delta, r)$ l'ensemble des itemsets δ-libres, alors les éléments de $\mathcal{B}d^-(\mathbb{F}(\gamma, \delta, r)) \cap \mathbb{F}(\delta, r)$ sont des itemsets non fréquents δ-libres et les itemsets minimaux de cet ensemble nous permettent de déterminer si un itemset J est fréquent ou non. En effet, s'il existe I un itemset non fréquent δ-libre minimal tel que $I \subseteq J$, alors J n'est pas fréquent sinon on peut approximer la fréquence de J.

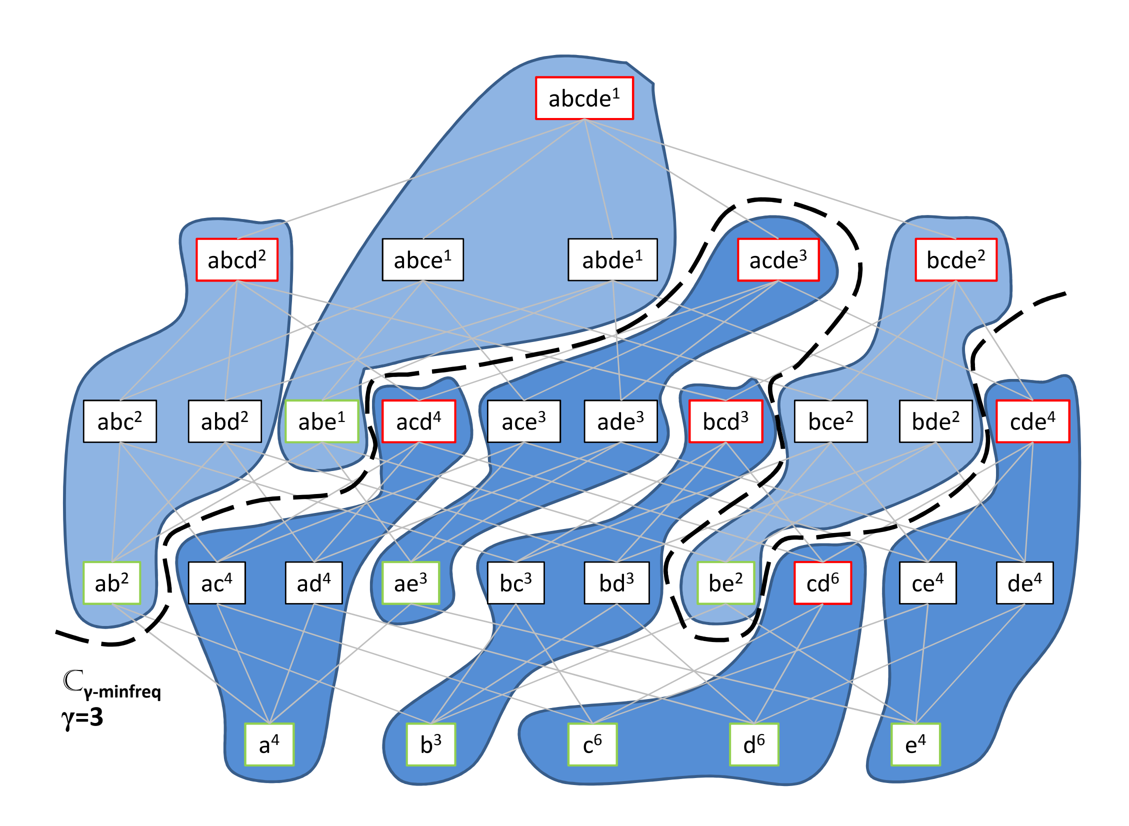


FIGURE 2.1 – Représentation des itemsets fermés, libres et des classes d'équivalence sous forme de treillis pour l'exemple de la table 2.1.

Lorsque $\delta = 0$, les itemsets 0-libres rejoignent la notion des itemsets clés (ou générateurs) [BTP+00]. Les notions d'itemset libre et fermé sont unifiées dans la notion de la classe d'équivalence de fermeture (ou encore appelé classe d'équivalence de support [BTP+00]). Dans une classe d'équivalence de support sont regroupés tous les itemsets

supportés par le même ensemble d'objets. Les itemsets d'une même classe d'équivalence ont bien sûr la même fréquence mais aussi la même unique fermeture. Ainsi, un itemset fermé est l'unique itemset maximal (selon $\subseteq$) d'une classe d'équivalence. Les itemsets 0-libres sont quant à eux les minimaux des classes d'équivalence – éventuellement plusieurs par classe d'équivalence.

En figure 2.1, nous représentons les classes d'équivalence pour l'exemple en table 2.1. En rouge les itemsets fermés, en vert les itemsets 0-libres, et en bleu clair les classes d'équivalence dont les éléments ne respectent pas la contrainte de fréquence minimum ($\gamma = 3$) représenté par la courbe en pointillé.

2.4 Autres représentations condensées

2.4.1 Les itemsets $\vee$-libres

Dans [BR01, BR03], les auteurs proposent une nouvelle représentation condensée des itemsets fréquents basée sur les itemsets $\vee$-libres. La notion d'itemset $\vee$-libre s'appuie sur la notion de $\vee$-règle. Intuitivement, une $\vee$-règle a un conséquent composé d'une disjonction d'attributs. Les auteurs les définissent formellement comme suit :

Définition 9 ($\vee$-règle et itemset $\vee$-libre) *Une $\vee$-règle est une règle de la forme $\pi : I \rightarrow a \vee b$, où $I \subseteq \mathcal{I}$ et $a, b \in \mathcal{I} \setminus I$. π est dite valide si et seulement si $supp(I, r) = \{t \in r \mid t \in supp(I \cup \{a\}) \vee supp(I \cup \{b\})\}$ – en d'autres termes, si toute transaction supportant I, supporte aussi a et/ou b.*
Un itemset $J \subseteq \mathcal{I}$ est un itemset $\vee$-libre si et seulement si il n'existe pas de $\vee$-règle valide $\pi : I \rightarrow a \vee b$ telle que $I \subset J$, $a, b \in J$ et $a \notin I$ et $b \notin I$.

De la définition d'une règle $\vee$-libre $\pi : I \rightarrow a \vee b$, on retire les égalités suivantes :

$$supp(I, r) = supp(I \cup \{a\}, r) + supp(I \cup \{b\}, r) - supp(I \cup \{a, b\}, r) \quad (2.1)$$
$$supp(I \cup \{a, b\}, r) = supp(I \cup \{a\}, r) + supp(I \cup \{b\}, r) - supp(I, r) \quad (2.2)$$

L'égalité 2.2 est équivalente à la validité de π et nous indique que nous pourrons déduire la fréquence de l'itemset $I \cup \{a, b\}$ grâce à la fréquence de trois itemsets de taille inférieure. Notons aussi que pour des raisons similaires à la propriété de δ-liberté, la propriété de $\vee$-liberté est anti-monotone.
Soit $\mathbb{F}(\gamma, \vee, r)$ l'ensemble des itemsets γ-fréquents $\vee$-libres de r. Pour pouvoir déduire si un itemset $J \subseteq \mathcal{I}$ est fréquent ou non, nous considérons en plus la bordure négative de $\mathbb{F}(\gamma, \vee, r)$ définie comme suit :

$$\mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r)) = \{I \in \mathcal{P}(\mathcal{I}) \setminus \mathbb{F}(\gamma, \vee, r) \mid \forall J \in \mathcal{P}(\mathcal{I}) : J \subset I \Rightarrow J \in \mathbb{F}(\gamma, \vee, r)\}$$

Ainsi, les ensembles $\mathbb{F}(\gamma, \vee, r)$, $\mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r))$ et les valeurs de fréquence de leurs éléments nous permettent de déterminer si un itemset J est fréquent et de donner sa fréquence. La démonstration se fait par induction sur la taille des itemsets : on suppose qu'à partir de $\mathbb{F}(\gamma, \vee, r)$, $\mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r))$ et des valeurs de fréquences de leurs éléments, on peut déduire la fréquence de tout itemset de taille inférieure ou égale à k.
Soit $J \subseteq \mathcal{I}$ un itemset tel que $|J| = k + 1$. Si $J \in \mathbb{F}(\gamma, \vee, r)$, alors il est fréquent et sa fréquence est connue. Si $J \notin \mathbb{F}(\gamma, \vee, r)$, alors il existe $I \subseteq J$ tel que $I \in \mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r))$. Considérons un tel itemset I. Bien sûr, si $I = J$, nous connaissons la fréquence de J. Dans le cas où $I \subset J$:

- soit I n'est pas γ-fréquent, auquel cas il n'y a aucune raison pour que J soit fréquent.
- soit I est γ-fréquent. Comme $I \in \mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r))$, I n'est pas $\vee$-libre et donc il existe $H \subset I$ et $a, b \in I \setminus H$ tels que $H \rightarrow a \vee b$ est une $\vee$-règle valide. En particulier, $I \setminus \{a, b\} \rightarrow a \vee b$ est valide. Donc nous avons l'égalité : $supp(I, r) = supp(I \setminus \{a\}, r) + supp(I \setminus \{b\}, r) - supp(I \setminus \{a, b\}, r)$. Puisque I est γ-fréquent, alors $I \setminus \{a\}, r$, $I \setminus \{b\}, r$ et $I \setminus \{a, b\}, r$ sont aussi fréquents et de taille inférieure à k. Par hypothèse d'induction, nous pouvons déduire leurs valeurs de fréquence. De plus, comme $I \subset J$, $J \setminus \{a, b\} \rightarrow a \vee b$ est aussi valide. Encore une fois, par hypothèse d'induction, si $J \setminus \{a\}, r$, $J \setminus \{b\}, r$ et $J \setminus \{a, b\}, r$ sont fréquents on peut déterminer leurs valeurs de fréquence dans ce cas. Et grâce à l'égalité $supp(J, r) = supp(J \setminus \{a\}, r) + supp(J \setminus \{b\}, r) - supp(J \setminus \{a, b\}, r)$, on pourra déterminer si J est fréquent et sa valeur de fréquence. Noter que J sera fréquent uniquement si $J \setminus \{a\}$, $J \setminus \{b\}$ et $J \setminus \{a, b\}$ sont fréquents.

Ainsi, $\mathbb{F}(\gamma, \vee, r)$ et $\mathcal{B}d^-(\mathbb{F}(\gamma, \vee, r))$ forment une représentation condensée des itemsets γ-fréquents.

Notons aussi la généralisation de ce concept : dans [KG02], les auteurs généralisent les itemsets $\vee$-libres en étendant la notion de $\vee$-règle à plusieurs disjonctions dans le conséquent. Ainsi, une $\vee$-règle généralisée est de la forme $\pi : I \rightarrow a_1 \vee \ldots \vee a_i \vee \ldots \vee a_n$. Et un itemset I sera $\vee$-libre généralisé si et seulement si pour tout $n > 0$, il n'existe pas de règle valide $\pi : I \setminus \{a_1, \ldots, a_i, \ldots, a_n\} \rightarrow a_1 \vee \ldots \vee a_i \vee \ldots \vee a_n$ telle que $\{a_1, \ldots, a_i, \ldots, a_n\} \subseteq I$.

2.4.2 Les itemsets non-dérivables

Introduit dans [CG02, CG07] comme une nouvelle représentation condensée des itemsets fréquents, les itemsets non-dérivables s'appuient sur un ensemble de règles de déduction afin de déduire des bornes pour la fréquence des itemsets. Puis, dans [CG03], les auteurs proposent une nouvelle représentation condensée basée sur les itemset k-libres [2].

2. Il faut bien faire la différence entre k-liberté et δ-liberté.

Règles de déduction et itemsets non-dérivables

Soit $I \subseteq \mathcal{I}$ un itemset. Les inéquations suivantes permettent de borner la fréquence de I en fonction de ses sous-ensembles $X \subseteq I$:

$$freq(I,r) \leq \sum_{X \subseteq J \subset I} (-1)^{|I \setminus J|+1} freq(J,r) \quad \text{si} \quad |I \setminus X| \quad \text{impair}$$
$$freq(I,r) \geq \sum_{X \subseteq J \subset I} (-1)^{|I \setminus J|+1} freq(J,r) \quad \text{si} \quad |I \setminus X| \quad \text{pair}$$

Par la suite, nous appellerons cette règle $\mathcal{R}_X(I)$. Selon la taille de I et de son ensemble X, la borne sera une borne supérieure (lorsque $|I \setminus X|$ est impair) ou une borne inférieure (lorsque $|I \setminus X|$ pair). Ainsi, si nous disposons de la fréquence de tous les sous-ensembles de I, alors nous pouvons déduire plusieurs bornes inférieures et supérieures pour la fréquence de I en utilisant $\mathcal{R}_X(I)$ pour tout $X \subseteq I$.

Notons $LB(I)$ la plus petite borne supérieure de I et $UB(I)$ sa plus grande borne inférieure. En pratique, il arrive souvent que $LB(I) = UB(I)$. Lorsque c'est le cas, I est un itemset dérivable puisqu'on connaît son fréquence : $freq(I,r) = LB(I) = UB(I)$. Les itemsets non-dérivables sont donc les itemsets I' tels que $LB(I') \neq UB(I')$. On peut démontrer que la propriété de non-dérivabilité est anti-monotone par rapport à la spécialisation des attributs [CG02]. Ainsi, si I est dérivable, alors ces sur-ensembles le sont aussi.
Une autre propriété intéressante des itemsets non-dérivables est qu'ils ne sont pas très grand – i.e. ils ne composés de peu d'attributs. En effet, soit l'intervalle $w(I) = UB(I) - LB(I)$. Les auteurs [CG02] montrent que $w(I)$ décroît exponentiellement par rapport à $|I|$. Ainsi, nous avons $w(I \cup \{i\} \leq w(I)/2$ pour tout itemset I et attribut $i \notin I$. Comme les $w(I)$ ne peuvent être divisés en deux qu'un "nombre logarithmique de fois", les itemsets non-dérivables ne seront pas très grands.
D'autre part, la taille d'une règle $\mathcal{R}_X(I)$ (i.e. le nombre de terme dans la somme des inéquations) augmente exponentiellement avec $|I \setminus X|$, appelé profondeur de $\mathcal{R}_X(I)$. Calculer toutes les règles peut paraître compliqué mais en pratique seules les règles de faible profondeur sont utilisées. Dans la suite, $LB_k(I)$ et $UB_k(I)$ dénoteront la plus grande borne inférieure et la plus petite borne supérieure pour I obtenues par l'évaluation de règles de profondeur inférieure ou égale à k. Ainsi, l'intervalle $[LB_k(I); UB_k(I)]$ est défini par les bornes calculées grâce à l'ensemble de règles $\{\mathcal{R}_X(I) \mid X \subseteq I \wedge |I \setminus X| \leq k\}$.

Les itemsets non-dérivables en tant que représentation condensée

Dans [CG02], les auteurs définissent une représentation condensée des itemsets fréquents $NDIRep$ basée sur les itemsets dérivables et les règles de déduction de la manière

suivante :

$$NDIRep(r,\gamma) = \{(I, freq(I,r)) \mid freq(I,r) \geq \gamma \wedge LB(I) \neq UB(I)\}$$

Grâce à $NDIRep$, pour tout itemset $I \subseteq \mathcal{I}$, il est possible de savoir si I est fréquent ou non, et s'il est fréquent, il est possible de déterminer sa valeur de fréquence. Considérons un itemset $I \notin NDIRep$. I est soit infréquent, soit dérivable (ou encore les deux). Après calcul de $LB(I)$ et $UB(I)$, si $LB(I) \neq UB(I)$ alors I n'est pas fréquent (sinon I appartiendrait à $NDIRep$). Si $LB(I) = UB(I)$, alors I est dérivable et $freq(I,r) = LB(I) = UB(I)$. Toutefois, pour calculer les bornes pour I, il est nécessaire de connaître la fréquence de tous les sous-ensembles de I. Premièrement, nous calculons les bornes des sous-ensembles de I qui sont dans la bordure négative de $NDIRep$ définie comme suit :

$$\mathcal{B}d^-(NDIRep(r,\gamma)) = \{I \in \mathcal{P}(\mathcal{I}) \backslash NDIRep(r,\gamma) \mid \forall J \in \mathcal{P}(\mathcal{I}) : J \subset I \Rightarrow J \in NDIRep(r,\gamma)\}$$

Si l'un d'eux est infréquent alors I sera infréquent aussi. Dans le cas contraire, nous connaissons les valeurs de fréquence de tous les sous-ensembles de I qui sont dans $\mathcal{B}d^-(NDIRep(r,\gamma))$. De la même manière, nous pouvons calculer les bornes pour les sous-ensembles de I qui sont juste au-dessus de $\mathcal{B}d^-(NDIRep(r,\gamma))$; et ainsi de suite jusqu'à ce que les fréquences de tous les sous-ensembles de I soient connus ou qu'un des sous-ensembles soit infréquent.

Extension de la non-dérivabilité et unification : itemsets k-libres

Définition 10 (Itemset k-libre) *Un itemset I est k-libre si $freq(I,r) \neq LB_k(I)$ et $freq(I,r) \neq UB_k(I)$. Un itemset sera ∞-libre si $freq(I,r) \neq LB(I)$ et $freq(I,r) \neq UB(I)$.*

Dans [CG03], les auteurs montrent que l'ensemble des itemsets γ-fréquents k-libres $\mathbb{F}(k,r)$ (et leurs valeurs de fréquence) agrémenté de la bordure suivante :

$$\{(J, freq(J,r)) \mid \forall j \in J : J \setminus \{j\} \in \mathbb{F}(k,r)\}$$

forme aussi une représentation condensée des itemsets γ-fréquents.

D'autre part, dans [CG03], les auteurs présentent les itemsets k-libres comme unificateurs des notions de 0-liberté (δ-liberté avec $\delta = 0$), $\vee$-liberté et non-dérivabilité (voit propriété suivante).

Propriété 1 *Soit $I \subseteq \mathcal{I}$ un itemset fréquent.*

– *I libre (i.e. δ-libre et $\delta = 0$) si et seulement si I est 1-free ($k = 1$).*

– *I est $\vee$-libre si et seulement si I est 2-libre ($k = 2$).*
– *I est $\vee$-libre généralisé si et seulement si I est ∞-libre.*

Cette nouvelle représentation condensée par les k-libres permet aussi de classer les différentes représentations par taille et par complexité (pour calculer la représentation et déduire les valeurs de fréquence) en fonction de la valeur de k. En effet, plus k est grand, plus la représentation est concise mais plus elle devient complexe. Dans la suite, nous discutons des principales applications des représentations condensées exposées.

2.4.3 Applications et discussion

En raison de la définition même des représentations condensées, toutes applications ayant besoin des itemsets fréquents y trouvent leur compte. En effet, les itemsets fréquents sont générés plus rapidement à partir des représentations condensées dans les contextes difficiles (e.g. données denses). Cependant, utiliser les représentations ne changent pas le nombre d'itemsets fréquents. Ainsi, dans des problèmes très difficiles, la taille même de l'ensemble complet des itemsets fréquents peut faire échouer le processus de regénération. C'est pourquoi certains chercheurs se sont intéressés à dériver des motifs pertinents (e.g. des règles d'associations) directement à partir des représentations condensées – sans passer par la collection complète (e.g. [Zak00, GMT05]).

Les représentations condensées qui ont attiré le plus d'applications diverse sont celles basées sur les propriétés de fermeture (i.e. les itemsets 0-libres et les itemsets fermés). Par exemple, les itemsets fermés dans des données d'expression des gènes représentent de réels groupes de synexpression [BRBR05]. Dans la suite de l'état de l'art, nous nous intéressons plus particulièrement aux multiples applications utilisant les itemsets δ-libres.

2.5 Usage multiple des itemsets δ-libres

Premièrement introduit en tant que représentation condensée (approximative) des itemsets fréquents, au fil des années qui suivirent, les itemsets δ-libres se sont trouvés de multiples usages dans diverses tâches de fouille de données :

– L'application la plus connue des itemsets δ-libres est certainement la dérivation de règles d'association δ-fortes qui se montrent plus pertinentes que les règles d'association classiques dans les données denses et fortement corrélées.
– Par delà les règles δ-fortes, les itemsets δ-libres ont aussi servi comme base pour construire un nouveau type de motif tolérant aux fautes (les δ-bi-ensembles [BPRB06]) qui se révèle pertinent dans les bases de données bruitées.

- Dans [PB05, PRB06], les δ-bi-ensembles sont aussi utilisés pour fournir une description de groupements d'objets et d'items issus de tâches de co-classification.
- Enfin, deux approches de classification associative [CB02, BC04] basées sur les itemsets δ-libres ont aussi vu le jour.

Dans cette section, nous passons en revue les principaux usages des itemsets δ-libres en fouille de données, avant d'ouvrir sur un nouvel usage pour la classification supervisée que nous présentons comme contribution au chapitre 3.

2.5.1 Règles d'association δ-fortes

De part la définition des itemsets δ-libres, il est clair que les notions d'itemset δ-libre et de règle δ-forte sont liés. D'un point de vue technique, les règles δ-fortes peuvent être construites à partir des itemsets δ-libres qui sont en fait les corps des règles δ-fortes [BBR03]. Les itemsets δ-libres sont aussi liés aux notions de presque-fermeture ou (δ-fermeture [BB00]) et de fermeture lorsque $\delta = 0$. En effet, lorsque $\delta = 0$, un itemset (0)-libre I et sa fermeture $J = cl(I, r)$ ayant le même support, il existe donc une règle forte (exacte) entre I et chacun des éléments de sa fermeture, i.e. $I \rightarrow a$ où $a \in J \setminus I$. Lorsque $\delta > 0$, on considère que la δ-fermeture $cl_\delta(I, r)$ d'un itemset δ-libre I est l'ensemble des attributs $b \in \mathcal{I}$ tels que $freq(I, r) - freq(I \cup \{b\}, r) \leq \delta$. Ainsi, il existe une règle δ-forte entre I et chacun des éléments de sa δ-fermeture, i.e. $I \rightarrow^\delta b$ où $b \in cl_\delta(I) \setminus I$. Notons que pour des valeurs faibles de δ par rapport à γ, les règles δ-fortes extraites commettent peu d'erreurs et sont alors de forte confiance.

Considérons l'exemple de base de données binaires de la table 2.1. Pour un seuil de fréquence minimum $\gamma = 3$ et un seuil d'erreurs maximum $\delta = 1$, l'ensemble des itemsets γ-fréquents δ-libres est : a, b et e. Leur δ-fermeture respective associée au gap de fréquence entre δ-libre et l'élément de la δ-fermeture est $\{c(0), d(0), e(-1)\}$, $\{a(-1), c(0), d(0), e(-1)\}$ et $\{a(-1), c(0), d(0)\}$. Par exemple, $a \rightarrow e$ est une règle 1-forte de confiance 0,75 et $a \rightarrow c$ est une règle de confiance 1.

L'extraction de règles δ-fortes apparaît comme une alternative à l'extraction de règles d'association classiques qui peut devenir impossible dans des contextes où le nombre d'itemsets fréquents est gigantesque (e.g. lorsque le seuil de fréquence est très bas et/ou le nombre d'attributs est relativement élevé et/ou les données sont très corrélées). En effet, l'approche classique requiert la recherche et le calcul de la fréquence d'au moins chaque itemset fréquent pour pouvoir ensuite générer des règles d'association valides (i.e. de fréquence et confiance dépassant des seuils donnés).
De plus, même lorsqu'il est possible de générer l'ensemble des règles valides, certaines de ces règles sont redondantes. On dit que $\pi_2 : I_2 \rightarrow J_2$ est redondante par rapport à $\pi_1 : I_1 \rightarrow J_1$, si π_1 et π_2 sont deux règles fréquentes de confiance proche et $I_1 \subseteq I_2$ et $J_2 \subseteq J_1$. Ainsi, on préférera π_1 à π_2 car π_1 est plus générale que π_2.
La notion de règle δ-forte propose une solution à ces deux problèmes. D'abord, au lieu

de s'appuyer sur l'ensemble des itemsets fréquents, on considère un sous-ensemble, les itemsets δ-libres qui seront les corps des futures règles. L'extraction des itemsets fréquents δ-libres supportent mieux les contextes difficiles énoncés plus haut. Puis, les conséquents de règles sont formés à partir de la δ-fermeture $cl_\delta(I, r)$ des itemsets δ-libres I.

- Lorsque $\delta > 0$, on peut approximer la fréquence de $I \cup J$ où $J \in cl_\delta(I, r) \setminus I$ ainsi que la confiance de $\pi : I \rightarrow cl_\delta(I, r) \setminus I$ et déduire si π est de fréquence et confiance suffisantes – on choisira les ensembles J maximaux comme conséquents de règles.
- Lorsque $\delta = 0$, les itemsets $cl_\delta(I, r) \setminus I$ constituent les conséquents de nos règles.

De cette manière, on obtient des règles dites maximales – $\pi : I \rightarrow J$ est dite maximale si il n'existe pas d'autre règle fréquente $\pi' : H \rightarrow K$ de confiance proche telle que $H \subseteq I$ et $J \subseteq K$. Dans cette direction, dans [BBJ+02], les auteurs appliquent l'extraction de règles δ-fortes ($\delta = 0$) à l'analyse de donnés d'expression de gènes humains [3].

2.5.2 Motifs tolérants aux erreurs

L'analyse de concepts formels [Wil82, GSW05] est une approche de découverte de connaissances qui a été très étudiée dans les bases de données binaires. Intuitivement, un concept formel est un rectangle maximal de 1 dans r. Par exemple, dans la table 2.1, le rectangle formé par le bi-ensemble des objets t_1, t_4 et des attributs b, d, c, e est un concept formel. Ainsi un concept formel associe un ensemble maximal d'objets à un ensemble maximal d'attributs. Dans la littérature, plusieurs algorithmes ont été proposés pour extraire l'ensemble des concepts formels (voir [KO02] pour une vue d'ensemble). Par construction, un concept formel est composé d'un ensemble fermé d'objets et d'un ensemble fermé d'attributs – tous deux liés par une connexion de Galois. Les récents challenges [GZ03, BGZ04] sur le calcul des itemsets fréquents fermés ont apporté des avancées non-négligeables pour la découverte de concepts formels autant en terme de performance via de nouveaux algorithmes qu'en terme de nouveaux domaines d'applications.

Toutefois, l'association entre les objets et les attributs induite par un concept formel est bien souvent trop forte dans des cas réels où les données sont imparfaites. C'est-à-dire que même si l'extraction complète reste faisable, c'est le post-traitement puis l'interprétation des résultats qui devient vite fastidieux voire impossible. Car, bien sûr, dans les données bruitées, le nombre de concepts formels peut être gigantesque, mais surtout beaucoup d'entre eux ne sont tout simplement pas pertinents. Ces limites ont motivées plusieurs travaux de recherche qui ont amené à étendre la notion de concept formel pour les données bruitées – le but étant de découvrir des rectangles contenant des imperfections, i.e. des bi-ensembles denses en valeur 1 mais contenant aussi un nombre de 0 contrôlé.

Les premières approches pour ce problème ont donné naissance à deux nouveaux motifs : les `CBS` (Consistent Bi-Set [BRB05b]) et les `DRBS` (Dense and Relevant Bi-

3. `SAGE` : Serial Analysis of Gene Expression.

Set [BRB05a]). CBS et DRBS sont définis avec des contraintes de consistance sur chaque ligne et chaque colonne à l'intérieur et à l'extérieur du bi-ensemble. Plus précisément, chaque objet (respectivement chaque attribut) hors du bi-ensemble doit contenir moins d'attributs (respectivement moins d'objets) qu'à l'intérieur du bi-ensemble. De telles contraintes rendent l'extraction de l'ensemble des CBS (ou DRBS) très coûteuse voire impossible dans de grandes bases de données.

Voulant adoucir ces contraintes, dans [BPRB06], les auteurs proposent d'utiliser un nouveau type de motif basé sur les itemsets δ-libres et leur δ-fermeture : les δ-bi-ensembles dits FBS [4]. Tout d'abord introduit dans [PB05] comme motif local pour la caractérisation de bi-regroupements issus de solutions de co-classification, les δ-bi-ensembles sont définis comme suit :

Définition 11 *Soit (T, I) un bi-ensemble dans r, tel que $T \subseteq \mathcal{T}$ et $I \subseteq \mathcal{I}$. (T, I) est un δ-bi-ensemble si et seulement si I peut être décomposé en $I = I_1 \cup I_2$ où I_1 est un itemset δ-libre dans r, I_2 l'ensemble des éléments de la δ-fermeture de I_1 ($I_2 = cl_\delta(I_1, r) \setminus I_1$) et $T = Objets(I_1, r)$. Ainsi, un 0-bi-ensemble est un concept formel.*

Dans un δ-bi-ensemble (T, I), le nombre de 0 par colonne (attribut de la partie δ-fermeture I_2 de I) est limité (mais pas par ligne). Ainsi, il peut exister des lignes en dehors de (T, I) avec le même nombre de 0 qu'une ligne à l'intérieur de (T, I). Cette perte de consistence permet toutefois une extraction de l'ensemble des δ-bi-ensembles dans les grandes bases de données puisqu'il est possible d'extraire les itemsets δ-libres et leur δ-fermeture dans de tels contextes.
Dans des données synthétiques artificiellement bruitées, les expériences [BPRB06] montrent qu'il est préférable de considérer les FBS que les concepts formels pour obtenir des bi-ensembles pertinents. Non seulement parce que le nombre de FBS n'explose pas dans les données bruitées et ainsi le post-traitement reste envisageable ; mais surtout parce que les δ-bi-ensembles sont plus proches des motifs originaux (sans bruit) que les concepts formels.

2.5.3 Caractérisation de groupes

La classification (non-supervisée) est une tâche importante de la fouille de données. Le but est d'identifier de groupements d'objets et/ou d'attributs de manière à optimiser une fonction objective qui évalue la qualité des regroupements. Par exemple, il peut être bon de maximiser les similarités entre éléments d'un même regroupement et les différences entre éléments de regroupements différents. Dans la littérature, il existe beaucoup d'algorithmes qui fournissent des partitions pertinentes d'objets et/ou d'attributs. Toutefois,

4. FBS pour Free set based Bi-Set.

une des grandes faiblesses des approches existantes vient du fait que l'on ne peut pas caractériser de manière explicite les partitions générées par un algorithme. En effet, on aimerait pouvoir expliquer pourquoi tels objets et/ou tels attributs se retrouvent dans le même groupement.

Les approches de co-classification donnent un premier élément de réponse à ce sujet. En effet, les techniques de co-classification génèrent des bi-regroupements (ou bi-partitions, i.e. une application entre une partition d'objets et une partition d'attributs). Chaque bi-regroupement est donc composé d'un ensemble d'objets $T \subseteq \mathcal{T}$ et d'un ensemble d'attributs $I \subseteq \mathcal{I}$. Intuitivement, on peut interpréter un bi-regroupement T, I de la manière suivante : les attributs de I sont des caractéristiques des objets de T. Toutefois, cette première interprétation reste au niveau global de la bi-partition et finalement on passe à côté de potentielles associations fortes entre sous-ensembles d'objets et sous-ensembles d'attributs.

Pour pallier ce problème, dans [PB05, PRB06], les auteurs proposent de compléter les techniques de co-classification par une étape de caractérisation des bi-regroupements en utilisant un ensemble de motifs locaux – une collection de bi-ensembles. Dans les expériences, concepts formels et δ-bi-ensembles sont mis à l'épreuve. Ainsi, partant d'un ensemble de k regroupements d'objets $\{C_1^T, \ldots, C_k^T\}$ associés par application à k regroupements d'attributs $\{C_1^I, \ldots, C_k^I\}$ qui donne une première caractérisation globale, les auteurs proposent d'associer chaque bi-ensemble au bi-regroupement qui lui ressemble le plus. L'évaluation de cette ressemblance se fait via une fonction de similitude sim entre un bi-regroupement (C_k^T, C_k^I) et un bi-ensemble X, Y définie comme suit :

$$sim((X,Y),(C_k^T, C_k^I)) = \frac{|X \cap C_k^T| \cdot |Y \cap C_k^I|}{|X \cup C_k^T| \cdot |Y \cup C_k^I|}$$

Intuitivement, sim mesure le rapport entre l'aire d'intersection des deux rectangles ((X, Y) et (C_k^T, C_k^I)) et l'aire de leur union. Le bi-ensemble (X, Y) est associé au bi-regroupement (C_k^T, C_k^I) qui maximise sim. Puis, afin de ne garder que les bi-ensembles les plus pertinents, seuls ceux dont les taux d'exceptions relatifs aux objets $\epsilon_T(X, C_k^T)$ et aux colonnes $\epsilon_I(Y, C_k^I)$ inférieurs à certains seuils ε_T et ε_I sont retenus. Ces taux d'exceptions sont définis comme suit :

$$\epsilon_T(X, C_k^T) = \frac{|\{x_i \in X \mid x_i \notin C_k^T\}|}{|X|} \quad , \quad \epsilon_I(Y, C_k^I) = \frac{|\{y_i \in Y \mid y_i \notin C_k^I\}|}{|Y|}$$

Expérimentalement parlant, dans les données bruitées, les δ-bi-ensembles sont plus pertinents (i.e. plus robustes au bruit) pour la caractérisation que les concepts formels. De plus, l'ensemble des δ-bi-ensembles est significativement plus petit que l'ensemble des concepts formels, facilitant l'interprétation.

2.5.4 Classification supervisée

Dans le chapitre 1, nous avons vu que les motifs fréquents (e.g. règles d'association) se révèlent très utiles en classification supervisée. Un des problèmes majeurs des approches de classification associative vient de la difficulté de l'extraction complète de l'ensemble des itemsets fréquents pour dériver des règles d'association (e.g. règles fréquentes et confiantes). De plus, lorsque l'extraction de règles est malgré tout possible, beaucoup d'entre elles mènent à des conflits de règles, sont redondantes, produisent du sur-apprentissage et donc sont inutiles. Les efforts de recherche de ces dernières années sur les représentations condensées des itemsets fréquents ont redonné espoir à la classification associative autant en terme de performance d'extraction que pour traiter les problèmes de redondance.

Dans [BC01, CB02], les auteurs développent une première approche de classification associative basée sur les itemsets δ-libres et les règles δ-fortes. Dans cette approche le centre d'intérêt est les règles δ-fortes de la forme $\pi : I \rightarrow^{\delta} c$ qui concluent sur un attribut classe. Étant donné qu'une règle δ-forte a un nombre d'exceptions borné par δ, on peut déduire une borne inférieure de la confiance des règles δ-fortes construites à partir d'itemsets δ-libres. Ceci est exprimé dans la propriété suivante :

Propriété 2 *Soit $\pi : I \rightarrow^{\delta} c$ une règle δ-forte, telle que $I \subseteq \mathcal{I}$ est un itemset γ-fréquent δ-libre. Alors, $conf(\pi, r) \geq 1 - \gamma/\delta$.*

Il est clair que pour des petites valeurs de δ (par rapport à γ), les règles δ-fortes sont de forte confiance (proche de 1). En plus, de cette propriété, la notion de règle δ-forte offre aussi une propriété de corps minimal, identifiée comme un point-clé en classification associative. On définit les règles de corps minimal de la façon suivante :

Définition 12 (Règle de corps minimal) *Soit γ seuil de fréquence minimum et δ un entier (seuil d'exceptions maximum). Une règle $\pi : I \rightarrow c$ a un corps minimal s'il n'existe pas d'autre règle $(\gamma - \delta)$-fréquente $\pi' : J \rightarrow c$ telle que $J \subseteq I$ et $conf(\pi', r) \geq 1 - \gamma/\delta$.*

Ainsi, nous nous intéressons aux règles de corps minimal qui concluent sur un attribut classe c (avec un degré d'incertitude gouverné par δ). Dans [CB02], les auteurs démontrent la propriété suivante qui lie règles de corps minimal et itemsets δ-libres.

Propriété 3 *Soit γ seuil de fréquence minimum et δ un entier (seuil d'exceptions maximum) et $\pi : I \rightarrow c$ une règle de corps minimal, alors I est un itemset δ-libre.*

Toutefois, ce lien n'est en rien une équivalence. Il peut exister des règles δ-fortes dont le corps δ-libre n'est pas minimal. Par exemple, dans la base de données de l'exemple 2.1

et la figure 2.1, pour $\delta = 0$, nous avons $ae \rightarrow c$ et $a \rightarrow c$ deux règles δ-fortes avec ae et a deux itemsets δ-libres et $a \subseteq ae$. Il est tout de même possible avec les techniques d'extraction par niveaux de reconnaître la propriété de corps minimal et ainsi d'obtenir directement les règles de corps minimal.

Sous certaines conditions sur les valeurs de γ et δ, il a été démontré que l'ensemble des règles δ-fortes fait fi de certains types de conflits de classification. En effet, si $\delta < \lfloor \gamma/2 \rfloor$, l'ensemble des règles δ-fortes ne peut contenir deux règles $\pi : I \rightarrow c_i$ et $\pi' : J \rightarrow c_j$ telles que $I' \subseteq I$ – de cette manière on évite des conflits dits d'inclusion de corps de règles.

Ainsi, l'ensemble des règles δ-fortes de caractérisation de classes sont les règles δ-fortes de corps minimal qui concluent sur un attribut classe et qui ne génèrent pas de conflits de classification. Cet ensemble est le centre du classifieur développé dans [BC01, CB02] et appliqué à la prédiction dans des données sur le cancer.

Plus tard, dans [BC04], les auteurs exploitent les propriétés de condensation et de non-redondance des itemsets 0-libres pour définir un ensemble de règles ε*ssentielles*, i.e. un ensemble minimal de règles ayant le même pouvoir de classification que l'ensemble complet de règles de classification. Par définition, une règle $\pi : I \rightarrow c$ est essentielle si et seulement si I est 0-libre. Cette formalisation des règles essentielles permet un gain en terme de performance à la fois dans la phase d'extraction des règles et dans la phase de post-traitement dans les techniques de classification associatives telles que `CBA` , `CMAR` .

2.6 Discussion

Enfin, la figure 2.2 résume bien le schéma de pensée de certains chercheurs qui est aussi le notre : comme les principales tâches de fouille de données s'appuient sur les itemsets fréquents, beaucoup de travaux de recherche se sont tout d'abord attaqués à l'extraction d'itemsets fréquents. Dans certains cas, face à la complexité du problème, l'extraction de la totalité des itemsets fréquents reste impossible, ou alors le résultat contient beaucoup de redondance et le post-traitement devient fastidieux. Les représentations condensées apportent des réponses à ses deux problèmes. Il est ainsi possible d'extraire une partie représentative des itemsets fréquents à partir de laquelle la génération de tous les autres itemsets fréquents via des mécanismes d'inférence est peu coûteuse. Toutefois, plutôt que de regénérer la totalité des itemsets fréquents, certains chercheurs se servent directement de l'ensemble représentatif (ou d'une partie) et de ses propriétés pour certaines tâches telles que la génération de règles d'association, la caractérisation de groupements, la classification associative, etc.

Dans le chapitre suivant, notre contribution adopte le schéma de la figure 2.2 et suit cette voie, puisque nous proposons une méthode de construction de descripteurs basée sur

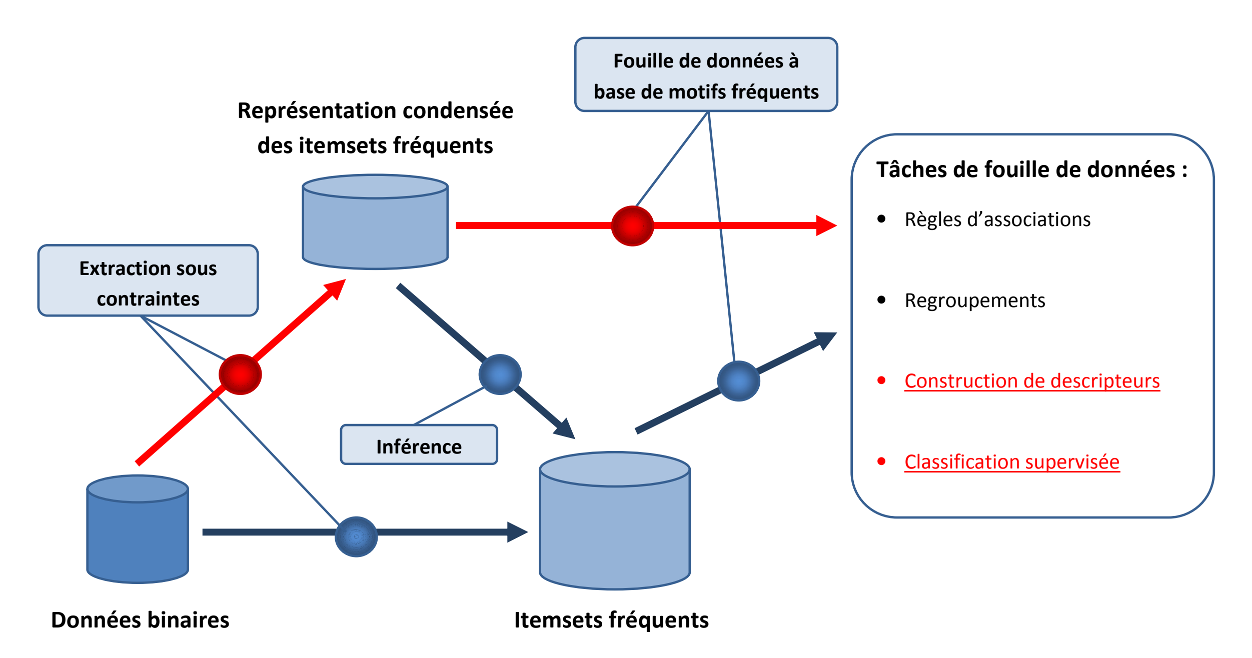


FIGURE 2.2 – Usage multiple des représentations d'itemsets fréquents

les itemsets δ-libres pour la classification supervisée, initiée dans [SLGB06].

Troisième partie

Contributions méthodologiques

Chapitre 3

Construction de descripteurs à base d'itemsets libres

Sommaire

3.1 Introduction

La construction de descripteurs est un des principaux thèmes de recherche dans les tâches de classification supervisée. A partir de l'ensemble des descripteurs originaux (attributs), le but est construire un ensemble de nouveaux descripteurs qui procurent une meilleure description des objets d'apprentissage afin d'améliorer les performances de prédiction des classifieurs appris. Dans la littérature, les premières approches se focalisent sur les arbres de décision. Originalement uni-variés (i.e. un seul attribut par noeud), les

arbres de décision deviennent alors multi-variés (i.e. plusieurs attributs sont pris en compte dans un noeud) par diverses méthodes : par exemple, un noeud peut être composé d'une combinaison linéaire d'attributs [UB90, BU95]. Si les résultats de précision sont meilleurs, l'arbre de décision perd grandement en interprétabilité.

Depuis les années 90 [AIS93], l'extraction de motifs fréquents est une des tâches phare de la fouille de données. Itemsets fréquents et règles d'association représentent certaines tendances ou associations entre sous-ensembles d'attributs dans les données. Dans le chapitre précédent nous avons vu que leur extraction est facilitée via les représentations condensées et, dans le chapitre présent, nous proposons une méthode de construction de nouveaux descripteurs basés sur les itemsets γ-fréquents δ-libres. Dans un premier temps, nous intégrons notre méthode dans la construction d'arbres de décision (dont nous rappelons le principe) afin d'obtenir un arbre de décision δ-`PDT` plus performants en terme de précision sans pour autant perdre en interprétabilité.

Puis par souci d'abstraction, nous généralisons notre approche de construction de descripteurs `FC` à plusieurs classifieurs et ce dans des contextes difficiles, i.e. aux attributs bruités. La présence de bruit dans les données peut avoir des effets désastreux sur la performance des classifieurs et donc sur la pertinence des décisions prises au moyen de ces modèles. Traiter ce problème lorsque le bruit affecte un attribut classe a été très étudié. Plusieurs approches ont été proposées pour, par exemple, l'élimination et la correction du bruit de classe ou encore la pondération des instances (e.g. [ZWC03, ZW04, RB07]). Au contraire, le problème du bruit d'attribut (non-classe) a été peu étudié. Il existe tout de même des travaux sur la modélisation et l'identification du bruit [KM03, ZW07] ainsi que des techniques de filtrage "pour nettoyer" les attributs bruités [ZW04, YWZ04]. Dans ce chapitre, nous proposons une méthode de construction de descripteurs robustes au bruit d'attributs sans pour autant éliminer les exemples bruités ni modifier les valeurs des attributs dans les données d'apprentissage. Nos descripteurs seront basés sur les itemsets δ-libres qui ont déjà fait leurs preuves pour la caractérisation de groupements dans les données bruitées (voir chapitre 2).

3.2 Arbre de décision à base de motifs

Dans cette section, nous décrivons notre approche de construction d'arbre de décision à base de motifs. Au lieu de considérer des attributs singletons comme noeuds de l'arbre, nous considérons des motifs plus pertinents : des disjonctions d'itemsets δ-libres. Avant tout, rappelons le principe de construction d'arbre de décision classique.

3.2.1 Principe des arbres de décision

La construction d'arbre de décision est une technique de classification supervisée très répandue. En effet, l'arbre de décision est un modèle prédictif très intuitif, applicable à de grosses bases de données et facile à interpréter. De plus il peut s'appliquer à des données binaires comme à des données catégorielles ou numériques. L'arbre de décision est un arbre au sens informatique du terme. Ses noeuds sont des tests sur des attributs de la base qui peuvent s'appliquer à tout objet de la base. Les réponses possibles aux tests des noeuds sont représentées par les labels des arcs qui sont issus des noeuds. Enfin, chaque feuille est étiquetée par une classe de la base. Ainsi pour prédire la classe d'un nouvel exemple t, il suffit de partir de la racine de l'arbre, de descendre dans l'arbre en suivant les arcs issus des noeuds tests que t vérifie, jusqu'à une feuille qui indique la classe à prédire.

Bien qu'il existe plusieurs algorithmes de construction d'arbre de décision, on peut généraliser ces algorithmes autour de trois points clés : *(i)* décider si un noeud est terminal (i.e. une feuille), *(ii)* sélectionner un test à associer à un noeud non terminal, *(iii)* affecter une classe à une feuille. Nous formalisons la construction générique d'un arbre de décision par l'algorithme 6. L'arbre est initialisé à la racine vide qui est le noeud courant (ligne 2). Si le noeud courant est terminal, alors on lui affecte une classe (ligne 6). Sinon successivement, on choisit le meilleur attribut $a \in \mathcal{I}$ qui maximise une certaine mesure d'intérêt discriminante pour l'attribut classe (ligne 8). Cet attribut sert de noeud test pour le noeud courant. Puis les données sont segmentées selon les valeurs de l'attribut a et on applique le même processus aux différents segments [1] créés (ligne 9) jusqu'à obtenir un noeud terminal.
D'autre part, selon les méthodes, il est possible d'élaguer les arbres (i.e. éliminer des sous-arbres) après construction afin de diminuer les erreurs commises par l'arbre et réduire les effets de sur-apprentissage dus à une sur-spécialisation de l'arbre construit.

`CART` [BFOS84], `ID3` [Qui86], et son extension `C4.5` [Qui93] sont les méthodes de construction d'arbres de décision les plus connues. Ils diffèrent de part leur façon de traiter les trois points clés cités précédemment mais aussi de leur méthode d'élagage après construction. Dans ce mémoire, nous nous intéressons plus particulièrement à `C4.5`. *(i)* `C4.5` décide qu'un noeud est terminal si tous les objets associés au noeud sont de la même classe ou s'il n'existe pas un nouvel attribut test pour lequel au moins deux branches sont associées à plus de n objets (par défaut $n = 2$). *(ii)* Pour choisir l'attribut test qui constitue un noeud, `C4.5` calcule le gain d'information pondéré (Gain Ratio) basé sur l'entropie dans le sous-ensemble de données courant r' pour chaque attribut a de la

1. Noter que dans le nombre de branches (dues aux segmentations) issues d'un noeud dépend de l'arité de l'attribut test du noeud. Ainsi dans le cas d'attributs binaires les données seront segmentées en deux parties, celle qui vérifie la propriété a et le reste.

Algorithme 6 : Algorithme générique de construction d'arbre de décision

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ une base de données,
$\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ l'ensemble des classes
Sortie : AD l'arbre de décision résultat

```
begin
    Initialisation de AD à l'arbre vide;
    Le noeud courant est la racine de l'arbre vide;
    repeat
        if Le noeud courant est terminal then
            Affecter une classe au noeud courant (i.e. feuille);
        else
            Sélectionner un attribut test;
            Créer le sous-arbre de AD associé à l'attribut test;
    until Toutes les feuilles sont étiquetées ;
end
```

manière suivante :

$$GainRatio(a, r') = \frac{IG(a, r')}{SI(a, r')}$$

où

$$IG(a, r') = E(r') - \sum_{a_j \in Dom(a)} freq_r(a_j, r') \times E(r'_{a_j})$$

IG est le gain d'information, $Dom(a)$ le domaine de valeurs pour l'attribut a, r'_{a_j} est la base de données restreinte aux objets ayant la valeur a_j pour l'attribut a et E est la fonction entropie définie comme suit :

$$E(r') = -\sum_{c_i = c_1}^{c_i = c_p} freq_r(c_i, r') \times \log_2(freq_r(c_i, r'))$$

Pour pondérer le gain d'information et favoriser les attributs binaires, `C4.5` utilise la fonction SplitInfo définie comme suit :

$$SI(a, r') = -\sum_{a_j \in Dom(a)} freq_r(a_j, r') \times \log_2(freq_r(a_j, r'))$$

Enfin, *(iii)*, `C4.5` attribue la classe majoritaire du sous-ensemble de données courant pour étiqueter les feuilles de l'arbre.

Considérons, l'exemple classique de base de données représenté par la table 3.1. Ici, $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ où $\mathcal{T} = \{t_1, \ldots, t_{14}\}$, $\mathcal{I} = \{outlook, temperature, humidity, windy, play\}$, $\mathcal{C} = \{yes, no\}$ et les relations de $\mathcal{R}$ entre objets et attributs sont représentées par le tableau

à deux dimensions en figure 3.1. Dans notre exemple, r est un ensemble de situations décrites par le temps qu'il fait et dans lesquelles on a joué au tennis ou non (*yes* ou *no*).

r		**Attributs**				**Classes**
		outlook	temperature	humidity	windy	play
Objets	t_1	sunny	85	85	FALSE	no
	t_2	sunny	80	90	TRUE	no
	t_3	overcast	83	86	FALSE	yes
	t_4	rainy	70	96	FALSE	yes
	t_5	rainy	68	80	FALSE	yes
	t_6	rainy	65	70	TRUE	no
	t_7	overcast	64	65	TRUE	yes
	t_8	sunny	72	95	FALSE	no
	t_9	sunny	69	70	FALSE	yes
	t_{10}	rainy	75	80	FALSE	yes
	t_{11}	sunny	75	70	TRUE	yes
	t_{12}	overcast	72	90	TRUE	yes
	t_{13}	overcast	81	75	FALSE	yes
	t_{14}	rainy	71	91	TRUE	no

TABLE 3.1 – Base de données exemple : `weather`.

Le problème est d'apprendre un modèle prédictif pour nous aider à décider si on va pouvoir jouer au tennis en fonction d'une nouvelle situation météorologique. Pour traiter les attributs continus comme *temperature* et *humidity*, `C4.5` utilise une méthode de discrétisation basée sur l'entropie. Il en résulte un attribut catégoriel dont chaque valeur décrit un intervalle de valeurs de l'attribut. Après application de l'algorithme `C4.5` grâce à la plateforme `WEKA` [WF05], l'arbre résultat est représenté en figure 3.1. Noter que, étant à la racine de l'arbre, l'attribut *outlook* est le plus discriminant pour la classe selon le gain ratio.

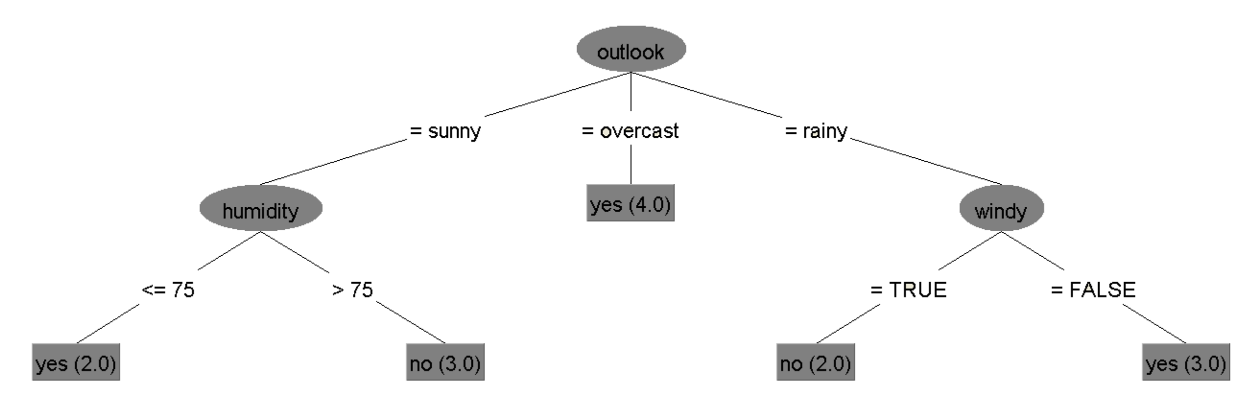


FIGURE 3.1 – Arbre de décision `C4.5` pour les données `weather`.

Nous avons vu que les itemsets fréquents peuvent contenir plus d'informations discriminantes pour la classe que les items singletons (attributs). L'idée est d'utiliser les itemsets fréquents discriminants pour la classe (au lieu des attributs simples) pour construire un nouveau type d'arbre de décision. Toutefois, l'ensemble des itemsets fréquents peut être très grand et contenir des éléments moins intéressants ou redondants par rapport à d'autres. Dans le chapitre 2, nous avons abordé le concept de représentation condensée de l'ensemble $\mathbb{F}$ des itemsets fréquents. Dans la suite, nous proposons une méthode de construction d'arbres de décision à base d'itemsets γ-fréquents δ-libres et de règles δ-fortes.

3.2.2 Règles δ-fortes et classes d'équivalence

Depuis [BTP$^+$00], il est maintenant commun de grouper les itemsets qui ont la même fermeture dans une classe d'équivalence de fermeture. En effet, ces itemsets décrivent les mêmes objets.

Définition 13 (Classe d'équivalence de fermeture ou de support) *Deux itemsets I et J sont dits équivalents par rapport à l'opérateur de fermeture cl (on note $I \sim_{cl} J$) si et seulement si $cl(I, r) = cl(J, r)$. Ainsi, une classe d'équivalence de fermeture (CEC[2]) est composé de tous les itemsets qui ont la même fermeture. Par définition de la fermeture, ils ont aussi le même support ($Objets(I, r) = Objets(J, r)$) et une CEC est aussi appelé classe d'équivalence de support.*

Chaque CEC contient un unique élément maximal (selon $\subseteq$) qui est un itemset clos (fermé). De plus, une CEC contient un ou plusieurs éléments minimaux qui sont des itemsets libres (appelés aussi itemsets clés dans [BTP$^+$00]). Cette formalisation utilisant les CECs est très intéressante pour dériver des règles d'association. En effet, la propriété 4 ci-dessous indique que l'on peut dériver une règle d'association forte (de confiance 1) entre un itemset libre et chaque élément de sa fermeture.

Propriété 4 *Soit F une classe d'équivalence de fermeture dans le contexte binaire r. Si $I \in E$ est un itemset δ-libre, alors $\forall j \in cl(I)$, $\pi : I \rightarrow j$ est une règle forte (de confiance 1).*

Ainsi, dans certains cas, une CEC pourra nous permettre de dériver des règles d'association concluant sur un attribut classe. En effet, il existe quatre cas typiques de CECs (voir figure 3.2).

Il est clair que la CEC du cas 1 dont l'itemset fermé ne contient pas d'attribut classe, ne nous permettra pas de dériver de règles fortes concluant sur une classe. Dans le cas 2,

2. CEC pour Closure Equivalence Class.

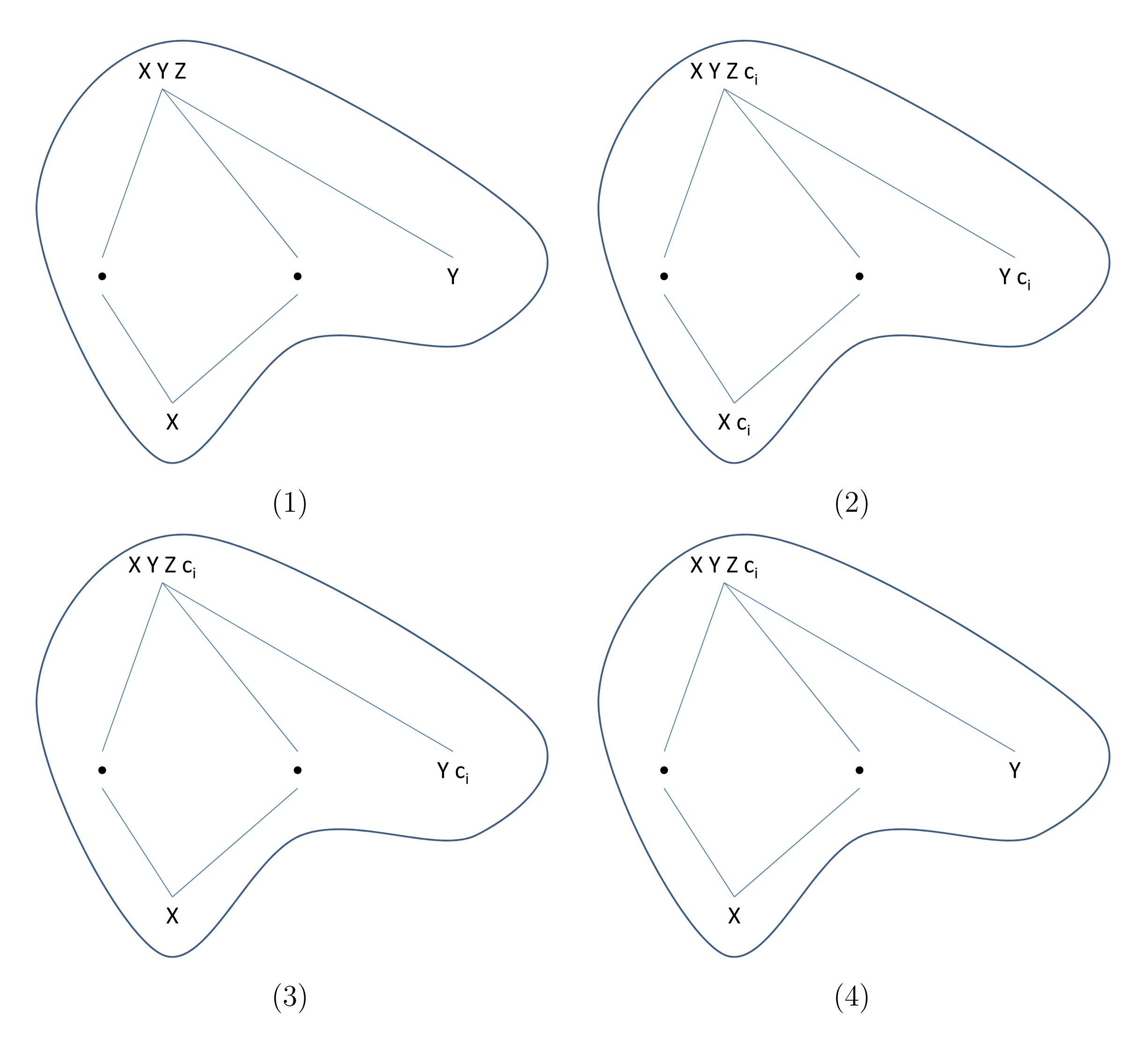


FIGURE 3.2 – Les 4 cas typiques de (δ)-CECs.

bien que l'itemset fermé contienne l'attribut classe c_i, tous les itemsets libres eux-aussi contiennent l'attribut classe – ce qui ne nous apprend rien sur la classe à part $Xc_i \rightarrow c_i$. Les cas 3 et 4 sont intéressants car dans la CEC, le fermé contient l'attribut classe et il existe un ou plusieurs libres qui ne contiennent pas la classe. Ainsi, on peut avoir $X \rightarrow c_i$ et/ou $Y \rightarrow c_i$ comme règles fortes (i.e. X et/ou Y sont des JEPs).

Bien sûr, il est possible de dériver d'autres règles dont le corps appartient aussi à la CEC et est un sur-ensemble d'un 0-libre. Toutefois, suivant l'intuition *"qui peut le plus peut le moins"*, nous choisirons les règles basées sur les minimaux (i.e. les libres) pour notre problème de classification supervisée. En effet, il suffira à un nouvel objet t de respecter les propriétés d'un itemset libre pour être considéré comme similaire aux objets décrits par tous les itemsets de la CEC dont fait partie l'itemset libre.

Toutefois, de tels motifs peuvent être rares dans des cas réels de données imparfaites ou légèrement bruitées. Pour introduire un peu de souplesse dans notre processus, nous parlerons ici de règles δ-fortes et de classes d'équivalence de δ-fermeture.

Définition 14 (Classe d'équivalence de δ-fermeture) *Deux itemsets I et J sont dits équivalents par rapport à l'opérateur de fermeture cl_δ (on note $I \sim_{cl_\delta} J$) si et seulement si $cl_\delta(I, r) = cl_\delta(J, r)$. Ainsi, une classe d'équivalence de δ-fermeture (δ-CEC) est composé de tous les itemsets qui ont la même δ-fermeture.*

Lorsque $\delta = 0$, les δ-CECs sont bien sûr des CECs. De la même manière qu'avec les CECs, les éléments minimaux des δ-CECs sont des δ-libres et il est possible de dériver des règles δ-fortes entre un itemset δ-libre et chaque élément de sa δ-fermeture. Encore une fois, le cas intéressant est lorsqu'un ou plusieurs itemsets δ-libres ne contiennent pas la classe qui est un élément de leur δ-fermeture. Ainsi, nous pouvons dériver des règles δ-fortes qui concluent sur un attribut classe.

Bien sur, le paramètre δ influe fortement sur la qualité des règles. Pour un γ donné, plus δ est grand, moins la règle δ-forte sera intéressante (règle dont le corps est un itemset γ-fréquent δ-libre). Selon certaines conditions sur les valeurs de γ et δ, nous allons nous assurer que les règles δ-fortes extraites sont bien discriminantes pour la classe.

Propriété 5 *Soit γ et δ deux entiers positifs et X un itemset γ-fréquent δ-libre dans r tel que $c \in cl_\delta(X, r)$ (où $c \in \mathcal{C}$ est un attribut classe).*
Alors $TA(X, r_c) > 1$ si $\delta \in [0; \gamma \times (1 - freq_r(c_i, r))[$ où $|r_{c_i}| \geq |r_{c_j}| \forall c_j \neq c_i$ (i.e. c_i est la classe majoritaire).

Ainsi, sous cette simple condition, les corps X de règles δ-fortes $\pi : X \rightarrow c$ extraites seront relativement plus fréquents dans r_c que dans le reste de la base. De plus, dans [CB02], les auteurs montrent que si $\delta \in [0; \gamma/2[$, alors l'ensemble $S_{\gamma,\delta}$ des règles δ-fortes extraites ne contient pas de conflits du type $X \rightarrow c_i$ et $Y \rightarrow c_j$ (pour $X \subseteq Y$ et $i \neq j$). Dans la suite, nous considérerons des ensembles $S_{\gamma,\delta}$ de règles δ-fortes disjonctives tel que γ et δ respectent cette condition ($0 \leq \delta < \gamma/2$) et la condition en propriété 5 et développons une méthode d'utilisation de cet ensemble pour construire un arbre de décision.

3.2.3 δ-PDT : un arbre de décision à base de règles δ-fortes

La construction d'arbre de décision à base de règles δ-fortes δ-`PDT`[3][GSB07] est décrite dans l'algorithme 7 et suit les principes de construction de l'algorithme `C4.5`. La différence

3. δ-`PDT` pour δ-Pattern based Decision Tree.

fondamentale se fait lors du choix des noeuds test où nous utiliserons une règle δ-forte au lieu d'un attribut. La mesure d'intérêt utilisée (e.g. le gain ratio) doit nous permettre de choisir le meilleure règle δ-forte discriminante pour les classes au lieu du meilleur attribut comme auparavant (ligne 6). Pour ce faire, nous proposons une adaptation triviale de calcul de la mesure d'intérêt pour des règles. En effet, pour un attribut $i \in \mathcal{I}$, le gain d'information exploite le nombre d'occurrences de i dans les différentes classes de données. En exploitant le nombre d'occurrences d'une règle δ-forte $\pi : I \rightarrow c_i$ dans les classes de données, nous pouvons calculer le gain d'information de π de la manière suivante :

$$IG(\pi, r) = E(r) - \Big(freq_r(I, r) \times E(r_\pi) + (1 - freq_r(I, r)) \times E(r_{\neg\pi})\Big)$$

Le splitinfo, étant basé sur la répartition des objets dans les classes en fonction des différentes valeurs de l'attribut courant, est étendu de la manière suivante :

$$SI(\pi, r) = -freq_r(I, r) \times \log_2(freq_r(I, r)) - (1 - freq_r(I, r)) \times \log_2(1 - freq_r(I, r))$$

Ainsi, le gain ratio est $GR(\pi, r) = IG(\pi, r)/SI(\pi, r)$. Noter que pour calculer GI etSI pour un motif les sommes sont réduites à deux termes puisqu'on considère qu'un motif apparaît ou n'apparaît pas dans un objet. Il en résulte des segmentations binaires et donc un arbre de décision binaire (lignes 7-12).
Noter que dans certaines δ-CECs, il est possible de dériver plusieurs règles δ-fortes qui concluent sur la même classe. Par définition d'une δ-CEC, ces règles concernent à peu près les mêmes objets de la base car leur corps est un itemset δ-libre d'une même δ-CEC. Un nouvel objet t sera considéré comme similaire aux objets décrits par les itemsets d'une δ-CEC, s'il respecte les propriétés d'un des itemsets δ-libres. Si la segmentation binaire se fait sur un des itemsets δ-libres (disons I) d'une δ-CEC, et que t respecte les propriétés d'un autre itemset J de cette δ-CEC mais pas celles de I, il ne sera pas considéré comme similaire aux objets de $Objets(I, r)$. Pour éviter ce désagrément, nous proposons d'utiliser des règles δ-fortes au corps disjonctif.

Définition 15 (Règles δ-fortes disjonctives) *Soit C une classe d'équivalence de δ-fermeture dont certains itemsets δ-libres $I_1, I_2, \ldots, I_k$ ne contiennent pas d'attribut classe et dont la δ-fermeture contient l'attribut classe c. L'unique règle δ-forte disjonctive de C qui conclut sur c est de la forme $\pi : I_1 \vee I_2 \vee \ldots \vee I_k \rightarrow c$.*

Noter que la fréquence du corps d'une telle règle δ-forte disjonctive est $freq(I_1 \vee I_2 \vee \ldots \vee I_k, r) = | \cup_{i=1}^{i=k} Objets(I_i, r)|$ – ce qui nous permet d'étendre facilement le calcul du gain d'information et du gain ratio pour les règles δ-fortes disjonctives. δ-`PDT` utilisera donc des règles disjonctives pour segmenter les données.

Enfin, δ-`PDT` est un algorithme récursif (lignes 9 et 12)et est donc lancé avec l'arbre vide et le contexte binaire r de départ.

Algorithme 7 : δ-`PDT` :Construction d'arbre de décision à base de motifs

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ l'ensemble des classes,
$S_{\gamma,\delta}$ l'ensemble des itemsets γ-fréquents δ-libres dont la δ-fermeture contient une classe,
AD l'arbre de décision courant

Sortie : PDT l'arbre de décision résultat

begin
- $NoeudCourant = AD.racine$;
- **if** `EstTerminal` *(NoeudCourant)* **then**
  - Affecter une classe au $NoeudCourant$;
- **else**
  - $a = \arg\max_{I \in S_{\gamma,\delta}} \{m(I, r)\}$ selon la mesure d'intérêt m;
  - $\mathcal{T}_{gauche} = \{t \in \mathcal{T} \mid t \in couv(a, r)\}$;
  - $r_{gauche} = \{\mathcal{T}_{gauche}, \mathcal{I}, \mathcal{R}'\}$;
  - $AD.filsGauche = \delta\text{-PDT}(r_{gauche}, \mathcal{C}, S_{\gamma,\delta}, \emptyset)$;
  - $\mathcal{T}_{droit} = \{t \in \mathcal{T} \mid t \notin couv(a, r)\}$;
  - $r_{droit} = \{\mathcal{T}_{droit}, \mathcal{I}, \mathcal{R}''\}$;
  - $AD.filsDroit = \delta\text{-PDT}(r_{droit}, \mathcal{C}, S_{\gamma,\delta}, \emptyset)$;
- $PDT \leftarrow AD$;

end

Revenons à notre exemple `weather`. La figure 3.2 rapporte une version discrétisée et binaire de la base exemple `weather`. Pour notre exemple, nous avons sélectionné les objets t_4 et t_6 comme objets test, le reste servant de base d'apprentissage. Considérons les paramètres de fréquence minimum et de nombre d'erreurs maximum permises suivants : $\gamma = 2$ et $\delta = 0$. L'ensemble des règles δ-fortes $\pi : I \rightarrow c_i$ concluant sur un attribut classe est alors $S_{2,0}$ (voir table 3.3).

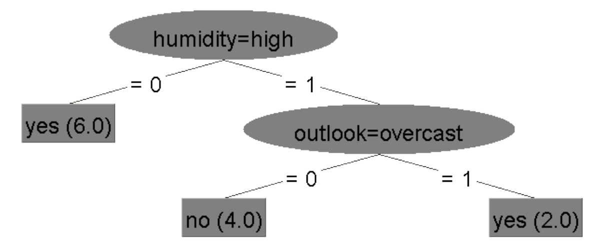
r		**Attributs**										**Classes**
		outlook			temperature			humidity		windy		play
		sunny	overcast	rainy	hot	mild	cool	high	normal	TRUE	FALSE	yes/no
Objets (Entraînement)	t_1	1	0	0	1	0	0	1	0	0	1	no
	t_2	1	0	0	1	0	0	1	0	1	0	no
	t_3	0	1	0	1	0	0	1	0	0	1	yes
	t_5	0	0	1	0	0	1	0	1	0	1	yes
	t_7	0	1	0	0	0	1	0	1	1	0	yes
	t_8	1	0	0	0	1	0	1	0	0	1	no
	t_9	1	0	0	0	0	1	0	1	0	1	yes
	t_{10}	0	0	1	0	1	0	0	1	0	1	yes
	t_{11}	1	0	0	0	1	0	0	1	1	0	yes
	t_{12}	0	1	0	0	1	0	1	0	1	0	yes
	t_{13}	0	1	0	1	0	0	0	1	0	1	yes
	t_{14}	0	0	1	0	1	0	1	0	1	0	no
Test	t_4	0	0	1	0	1	0	1	0	0	1	yes
	t_6	0	0	1	0	0	1	0	1	1	0	no

TABLE 3.2 – Base de données binaires : `weather`.

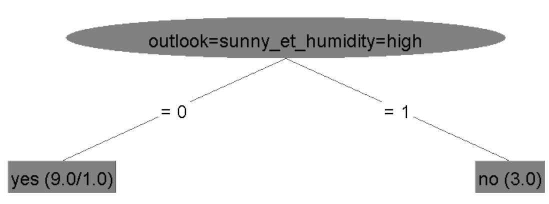
$S_{2,0}$	$freq(I, r)$
outlook=overcast → yes	4
temperature=cool → yes	3
humidity=normal → yes	6
outlook=sunny ∧ temperature=hot → no	2
outlook=sunny ∧ humidity=high → no	3
outlook=rainy ∧ windy=FALSE → yes	2

TABLE 3.3 – Liste des règles de $S_{2,0}$ pour la base `weather` binaire.

Puis, en utilisant notre algorithme 7 sur la base d'entraînement, δ-`PDT`(`weather_entrainement`, $\{yes, no\}, S_{2,0}, \emptyset$) nous permet de construire l'arbre de décision présenté en figure 3.3b. Nous le confrontons à l'arbre de décision `C4.5` construit sur les mêmes données d'apprentissage (voir figure 3.3a). Nous remarquons que les premières segmentations de δ-`PDT` et `C4.5` sont différentes. Il en résulte une séparation des données différentes. `C4.5` sépare les données selon l'attribut qui récolte le plus haut gain ratio $humidity = high$. Résultat de la séparation : ($6yes/0no$) et ($2yes/4no$). De son côté, δ-`PDT` utilise le gain ratio de la règle 0-forte π : $outlook = sunny \wedge humidity = high \rightarrow no$ pour segmenter les données. Résultat de la séparation : ($8yes/1no$) et ($0yes/2no$). Ainsi, pour `C4.5`, $IG(humidity = high, r) = 0.4592$ ($GR(humidity = high, r) = 0.4592$) et pour δ-`PDT`, $IG(\pi, r) = 0.5409$ ($GR(\pi, r) = 0.6667$). Selon IG et GR, π est plus discriminante que n'importe quel attribut singleton. Il en résulte une meilleure segmentation des données.



(a) Arbre de décision C4.5 appris sur la base d'entraînement.



(b) Arbre de décision à base de motifs appris sur la base d'entraînement.

FIGURE 3.3 – Arbres de décision sur `weather`

Considérons maintenant, les deux objets test (t_4, t_6). `C4.5` classe mal ces deux situations car pour t_4, pour les propriétés *humidity* = *high* et *outlook* $\neq$ *overcast*, `C4.5` indique *no* et pour t_6, pour la propriété *humidity* $\neq$ *high*, `C4.5` indique *yes*. Au contraire, avec une segmentation différente, δ-`PDT` permet de bien classer t_4. Avec cet exemple, nous avons une première idée des effets de l'utilisation de règles δ-fortes pour construire un arbre de décision. Toutefois, il parait clair que ce processus dépend fortement des paramètres γ (fréquence) et δ (nombre d'erreurs) qui contraignent l'extraction de $S_{\gamma,\delta}$. Dans la suite, nous étudions via des expériences les effets de γ et δ sur notre processus et nous le testons sur un plus large panel de bases de données.

3.2.4 Paramétrage du processus et validation

Impact de γ et δ

D'un côté, le dilemme lié au seuil de fréquence minimum γ est bien connu. Pour un seuil très petit, le nombre de règles dans $S_{\gamma,\delta}$ est considérable. Parmi ces règles, beaucoup sont peu utiles puisqu'elles sont très locales et ne concernent qu'un petit sous-ensemble de données. Ces règles auront une faible valeur de gain d'information et il en résultera une segmentation peu pertinente des données. Pour un seuil très grand, trop peu de règles seront générées pour pouvoir couvrir raisonnablement la base d'apprentissage, ce qui ne donne pas une bonne représentativité des données d'apprentissage. De plus, nous l'avons vu en figure 3, les itemsets très fréquents ont une valeur de gain d'information limitée ce qui implique une segmentation peu pertinente des données.
D'autre part, le nombre d'erreurs maximum autorisées δ influence aussi la qualité des règles extraites. En effet, pour $\delta = 0$, les règles de $S_{\gamma,\delta}$ sont des règles fortes (de confiance 1), ou encore, les itemsets γ-fréquents δ-libres (corps de règles) sont des JEPs. Malheureusement, si les données sont imparfaites (i.e. légèrement bruitées, et c'est souvent le cas), les JEPs peuvent être rares et l'ensemble des JEPs n'offrira pas une bonne couverture des

données. Évidemment, les très grandes valeurs de δ mènent à des règles peu discriminantes en raison du trop grand d'erreurs acceptées.

Le point clé semble être la couverture de la base d'apprentissage. Pour mieux comprendre les effets du paramétrage de notre processus sur la couverture, dans les graphiques de la figure 3.4, nous représentons le taux de couverture de la base d'apprentissage en fonction de γ et δ. Nous prenons ici comme exemples les bases `breast`, `cleve`, `heart`, `hepatic` du répertoire de base de données `UCI` [4]. Chaque courbe représente un seuil de fréquence γ. En ordonnée, le taux de couverture est représenté en fonction de δ (en abscisse). Comme attendu, pour δ petit, le taux de couverture est bas. Plus δ augmente, plus le taux de couverture augmente jusqu'à un point de stabilisation (proche ou égal à 100%). Nous pensons que ces points de saturation (notés δ_{opt}) sont importants et indiquent la valeur de δ à choisir pour l'extraction. En effet, pour $\delta < \delta_{opt}$, $S_{\gamma,\delta}$ ne couvre pas assez bien la base d'apprentissage et pour $\delta > \delta_{opt}$, $S_{\gamma,\delta}$ contient des règles moins pertinentes que $S_{\gamma,\delta_{opt}}$.

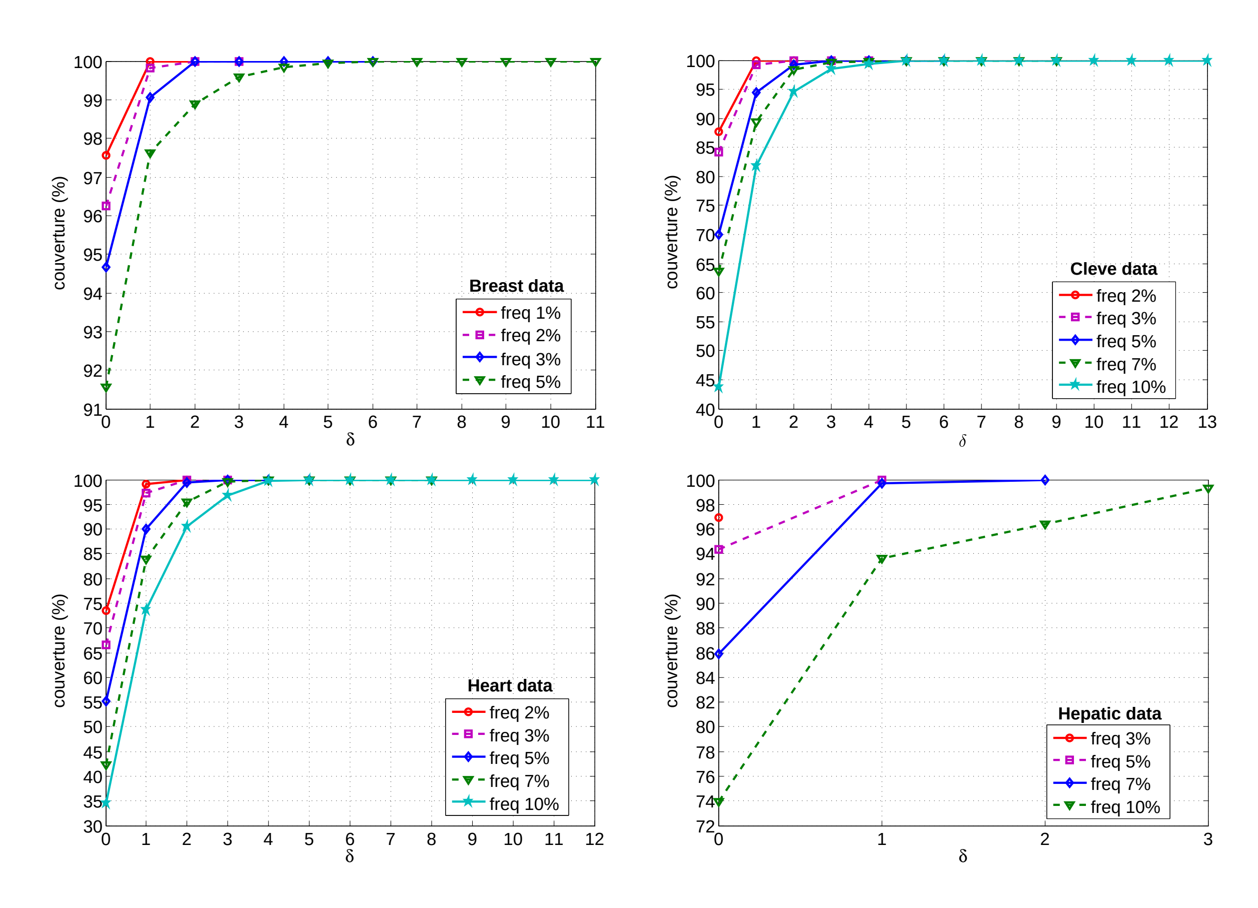


FIGURE 3.4 – Processus générique de construction de descripteurs à base de motifs

4. http://archive.ics.uci.edu/ml/ University of California, Irvine.

Résultats de précision et comparaison

Nous testons notre méthode δ-PDT sur onze bases de données UCI [AN07] et une base de données meningitis (voir les descriptions en annexe). Pour avoir une bonne estimation de la précision de δ-PDT, nous procédons à une validation croisée à 10 blocs stratifiés selon les classes (10-SCV) [5]. C'est-à-dire, chaque échantillon respecte à l'unité prêt la répartition des objets dans les différentes classes. Certaines bases UCI contiennent des attributs continus ; dans ce cas, on utilise une méthode de dicrétisation supervisée, basée sur l'entropie [FI93], sur les données d'apprentissage, puis les attributs discrets sont binarisés pour pouvoir extraire les règles δ-fortes. Le schéma de binarisation est ensuite reporté sur les données test.

Pour δ-PDT , les valeurs de fréquence relative des itemsets δ-libres testées varient entre 0% et 10% (excepté pour les données car pour lesquelles $freq_r \in [0\%; 0.8\%]$). Les valeurs de δ sont contraintes par γ afin que les itemsets γ-fréquents δ-libres soient émergents ($\rho > 1$). Ainsi, nous reportons deux types de résultats de précision : *(i)* le meilleur résultat de précision obtenu avec δ-PDT toutes combinaisons (γ, δ) confondues ; *(ii)* la moyenne sur les valeurs de γ des précisions pour $\delta = \delta_{opt}$.

Nous comparons les résultats de précision de δ-PDT avec C4.5 et d'autres classifieurs de la littérature : CBA , CPAR , SJEP-C. pour C4.5, nous utilisons l'implémentation J48 de la plateforme WEKA et la même discrétisation et validation croisée que pour δ-PDT . Pour CBA , nous utilisons la version disponible en ligne [6] avec les paramètres de fréquence et confiance des règles comme indiqués dans l'article original : 1% et 50%. Et pour les autres classifieurs nous reportons les résultats annoncés dans les articles originaux. Tous ces résultats de précision sont reportés dans la table 3.4. La meilleure précision obtenue pour une base de données est indiquée en gras.

On remarque qu'il existe souvent (7 fois sur 12) une combinaison de paramètres (γ, δ) pour laquelle δ-PDT obtient de meilleurs résultats de précision que ces concurrents. Dans le détail, avec la meilleure combinaison (γ, δ), δ-PDT l'emporte sur CBA (11/12), sur CPAR (8/12), sur SJEP-C (6/8) et enfin sur C4.5 (11/12) – ce qui est encourageant pour peu qu'on ait la bonne combinaison. Les résultats moyennés obtenus par δ-PDT avec les δ_{opt} sont moins percutants par rapport aux autres classifieurs. Toutefois, on voit tout de même que l'emploi des nouveaux descripteurs est bénéfique dans δ-PDT par rapport aux attributs originaux dans C4.5. En effet, (voir colonne moyenne), δ-PDT utilisant les δ_{opt} l'emporte 8 fois sur 12 face à C4.5. Enfin, la dernière ligne – moyenne pondérée des précisions sur toutes les bases – montre que δ-PDT obtient des résultats meilleurs que l'arbre de décision C4.5 original.

5. 10-SCV, en anglais pour 10-folds stratified cross-validation.
6. CBA http://www.comp.nus.edu.sg/~dm2/

Données	C4.5	CBA	CPAR	SJEPs	δ-PDT		
					moyenne	meilleur	$(freq_r\,;\delta)$
car	92.36	88.90	92.65	-	93.98	**94.10**	{(0.2 ;0)}
cleve	78.88	**83.83**	83.61	82.41	82.07	**83.83**	{(10 ;{3,4})}
heart	83.70	81.87	83.70	82.96	83.49	**85.93**	{(5 ;3)}
hepatic	82.58	81.82	84	83.33	82.74	**85.16**	{(10 ;2),(5 ;0)}
horse-colic	**85.05**	81.02	84.14	84.17	81.93	84.51	{(10 ;{12,13})}
iris	93.33	**95.33**	94.67	-	95.11	**95.33**	{(7 ;{2,3,4}),(5 ;{2,3,4}),(3 ;2),(2 ;2)}
labor	82.46	86.33	**91.17**	82.00	85.61	87.72	{(10 ;0),(7 ;0),(5 ;0)}
lymph	77.03	84.50	80.28	-	81.98	**86.49**	{(10 ;3)}
meningitis	94.83	91.79	91.52	-	92.25	**95.13**	{(3 ;2)}
sonar	79.81	79.81	84.07	**85.10**	79.33	81.25	{(10 ;5)}
vehicle	69.98	67.99	**73.3**	71.36	70.28	71.04	{(1 ;5)}
wine	96.07	94.96	**97.54**	95.63	96.63	97.19	{(10 ;6)}
Moyenne	84.67	84.85	86.72	83.37	85.45	**87.31**	-

TABLE 3.4 – Comparaison des précisions de δ-PDT

3.2.5 Discussion

Au vu de la validation expérimentale, δ-PDT est comparable en terme de précision aux classifieurs à base de motifs (CPAR , SJEP-C) et semble plus performant que l'arbre de décision original C4.5 et CBA . Ceci est dû aux nouveaux descripteurs discriminants pour la classe que nous avons construits à partir des itemsets γ-fréquents δ-libres (et qui deviennent les noeuds).
Depuis, d'autres méthodes ont vu le jour. Notons par exemple le travail de A. Zimmermmann [Zim08]. L'auteur propose une approche ET (Ensemble Trees) similaire pour utiliser le pouvoir discriminant d'ensemble de règles dans les arbres de décision – à la différence près qu'à chaque itération de la construction de l'arbre, un nouvel ensemble de k meilleures règles est extrait. Par rapport aux arbres originaux, la taille des ET est moindre et les résultats de précision des ET sont comparables.

Dans cette section, nous nous sommes restreints à la construction de descripteurs pour les arbres de décision. Dans la suite, nous allons plus loin dans la formalisation de la construction de descripteurs puis étendons notre méthode pour qu'elle soit applicable pour tout type de classifieurs – pas seulement aux arbres de décision. D'autre part, pour faire face au bruit potentiel dans les données, nous proposons une méthode de construction de descripteurs numériques plus "souples" (au lieu de descripteurs binaires) [GSB08, GSB09, SGB09].

3.3 Processus générique de construction de descripteurs

Avant de développer notre approche générique, nous la motivons par une étude précise des dernières approches de la littérature en classification à base de motifs fréquents utilisant les propriétés de fermeture.

Itemsets libres ou itemsets fermés

Dans la littérature, l'utilisation des représentations condensées basées sur les propriétés de fermeture a déjà été étudiée pour la classification supervisée. On peut différencier deux types d'approches :
- *(i)* après avoir enlevé l'attribut classe de la base de données, dans [LLW07], les auteurs proposent d'extraire les itemsets libres et les itemsets fermés pour préférer finalement les libres pour les problèmes de classification. De même, dans [BC04], les auteurs construisent des règles essentielles dont le corps est un itemset libre.

– *(ii)* après avoir partitionné la base de données selon l'attribut classe, dans [GKL06, GKL08], les itemsets fermés sont choisis pour caractériser les classes. De même, dans [CYHH07], les auteurs proposent de construire de nouveaux descripteurs à partir des itemsets fermés. Pour éclaircir cette divergence d'avis, nous détaillons les principes des diverses approches.

Approche classique. Dans la première approche, bien que la maximalité fait des itemsets fermés de bons candidats pour caractériser des données labellisées, cette même maximalité n'est pas efficace quand il s'agit de prédire. Ainsi, pour la prédiction, l'utilisation des itemsets libres sera plus judicieuse en raison de leur minimalité. Noter qu'en raison des propriétés de fermeture, tous les itemsets d'une même CEC couvrent exactement les mêmes objets de r et donc ont la même fréquence. Donc, tous les itemsets libres d'une CEC et leur itemset fermé correspondant sont équivalents par rapport à toute mesure d'intérêt basée sur la fréquence. Bien que dans ce cadre, libres et fermés sont équivalents, c'est lors de la phase de prédiction que le choix va se faire.
Considérons un nouvel objet t entrant qui est décrit exactement par l'itemset Y (i.e. $Items(t, r) = Y$). Supposons que $F \subseteq Y \subseteq cl(F, r)$, tel que F est un itemset libre et donc $F, Y, cl(F, r)$ font partie de la même CEC C_F. Ici, utiliser les libres ou les fermés pour prédire la classe de t ne conduira pas à la même décision. En effet, $Items(t, r) \supseteq F$ et t est couvert par la règle $F \rightarrow c$, alors que $Items(t, r) \not\supseteq cl(F, r)$ et n'est pas couvert par la règle $cl(F, r) \rightarrow c$. Suivant l'intuition *"Qui peut le plus peut le moins"*, les libres sont ici préférés aux fermés.

Approche par classe. Pour les approches par classe, considérons, sans perte de généralité, un problème à deux classes (c_1, c_2). Dans un tel contexte, l'équivalence entre libres et fermés par rapport aux mesures d'intérêt basées sur la fréquence ne tient plus. En effet, l'approche par classe suit l'intuition suivante : soit Y un itemset libre dans r_{c_1} et $X = cl(Y, r_{c_1})$ son fermé. L'itemset fermé X est considéré comme plus pertinent que Y (ou tout autre itemset de la CEC) pour caractériser c_1 car $Objets(X, r_{c_1}) = Objets(Y, r_{c_1})$ et $Objets(X, r_{c_2}) \subseteq Objets(Y, r_{c_2})$. Dans [CYHH07], les auteurs se contentent des itemsets fermés pour caractériser c_1. D'autre part, dans [GKL06], pour caractériser c_1, les auteurs choisissent les itemsets fermés $X = cl(X, r_{c_1})$ tels qu'il n'existe pas d'autre fermé $X' = cl(X', r)$ pour lequel $cl(X, r_{c_2}) = cl(X', r_{c_2})$. Bien sûr, dans certains cas, un itemset libre peut être équivalent à son fermé $X = cl(Y, r_{c_1})$, c'est-à-dire $Objets(X, r_{c_2}) = Objets(Y, r_{c_2})$. Dans ces cas-là, pour les mêmes raisons que précédemment, le libre sera préféré. Noter tout de même que dans cette approche, la pertinence des itemsets fermés ne permet pas d'éviter de générer des règles conflictuelles. En effet, il est possible d'avoir deux itemsets fermés X et Y pertinents pour c_1 et c_2 respectivement alors que $X \subseteq Y$.

Ainsi, le dilemme du choix entre libres et clos comme motifs pour la classification supervisée est principalement due à la façon d'extraire les motifs : extraire dans toute

la base ou par classe. De plus, comme les deux approches ne prennent pas en compte la distribution des classes dans les données, un post-traitement est nécessaire pour accéder aux motifs (libres ou fermés) discriminants. En effet, nous ne recherchons pas uniquement les propriétés de fermeture pour traiter la redondance mais nous voulons aussi exploiter les mesures d'intérêt pour obtenir des motifs discriminants. Pour prendre en compte la distribution des classes et éviter un post-traitement, nous proposons par la suite de garder l'attribut classe lors de la phase d'extraction et ainsi utiliser l'information contenue dans une CEC qui contient l'attribut classe. Comme précédemment, nous nous intéresserons aux δ-CECs, dont un δ-libre ne contient pas d'attribut classe et dont la δ-fermeture contient une classe (voir les cas 3 et 4 de la figure 3.2). Dans la suite nous montrons que les δ-CECs de notre approche contiennent plus d'informations que les motifs utilisés dans les autres approches classique ou par classe.

Informations contenues dans une δ-CEC

Considérons la table de contingence pour la règle de classification $\pi : X \rightarrow c$ en figure 3.5. Pour toutes les approches (classique, par classe et la notre), la distribution des classes (f_{*1} et f_{*0}) est connue et il en est de même pour le nombre d'objets de la base f_{**}. Toutefois, l'approche par classe ne nous donne directement accès qu'à la valeur f_{11} qui est la fréquence de l'itemset fermé X dans r_{c_1}. Par déduction, nous connaissons aussi f_{01}, mais nous ne savons rien de la fréquence de X dans le reste de la base (f_{10} et f_{00}).
Au contraire, l'approche classique nous renseigne directement sur la valeur de f_{1*} qui est la fréquence de l'itemset fermé (ou libre) X dans r. Par déduction, on connaît f_{0*} mais on ne sait rien sur la fréquence de X dans les différentes classes.
Enfin, dans notre approche, puisque $\pi \in S_{\gamma,\delta}$, X est γ-fréquent δ-libre et $c \in cl_\delta(X, r)$. Ainsi, f_{1*} la fréquence de X (notons $\gamma_X \geq \gamma$) et f_{01} le nombre d'erreurs commises (notons $\delta_X \leq \delta$) par π sont connues. Et par déduction, nous connaissons aussi toutes les autres valeurs de la table : f_{11} la fréquence de π dans r est $(\gamma_X - \delta_X)$; f_{01} et f_{00} sont respectivement $(|\mathcal{T}_c| - (\gamma_X - \delta_X))$ et $(|\mathcal{T}_{\bar{c}}| - \delta_X)$.

$\pi : X \rightarrow c$	c	$\bar{c}$	Σ
X	f_{11}	f_{10}	f_{1*}
$\bar{X}$	f_{01}	f_{00}	f_{0*}
Σ	f_{*1}	f_{*0}	f_{**}

FIGURE 3.5 – Table de contingence pour la règle de classification $\pi : X \rightarrow c$ qui conclut sur la classe c.

La connaissance des valeurs f_{11} et f_{10} que procure le concept de δ-CEC est très précieuse. En classification supervisée basée sur les motifs, la plupart des mesures d'intérêt basées sur la fréquence sont calculées à partir de ces deux valeurs : i.e. la fréquence du

corps d'une règle dans la classe qu'elle caractérise et sa fréquence dans le reste de la base. Les autres approches (classique ou par classe), ne disposant pas de ces deux valeurs, post-traitent l'ensemble de motifs. Dans la suite, nous montrons que le concept de δ-CEC et l'information qu'il contient permettent d'éviter cette phase.

Propriétés d'une CEC

Selon la formalisation de [CB02], $\pi : X \rightarrow c_i$ est une règle d'association δ-forte de caractérisation (δ-SCR [7]) si c_i est un attribut classe et le corps X est minimal. X est minimal s'il n'existe pas d'autre règle fréquente [8] $\pi' : Y \rightarrow c_i$ telle que $Y \subseteq X$ et $conf(\pi', r) \geq 1 - \gamma/\delta$.

Cette formalisation nous permet d'éviter des conflits de règles dans l'ensemble des δ-SCRs extraites sous de simples conditions sur les valeurs de γ et δ. En effet, si $\delta \in [0; \lfloor \gamma/2 \lfloor[$, nous n'avons pas de conflits de corps de règles dans $S_{\gamma,\delta}$, c'est-à-dire qu'il n'existe pas deux règles $\pi : X \rightarrow c_i$ et $\pi' : Y \rightarrow c_j$ telles que $j \neq i$ et $Y \subseteq X$.

Cependant, la définition de δ-SCR basée uniquement sur la mesure de confiance n'est pas suffisante pour les tâches de prédiction. Dans [BMS97], les auteurs montrent que même des règles $\pi : X \rightarrow c_i$ de forte confiance n'assurent pas que X est positivement corrélé avec c_i et donc peuvent induire en erreur. C'est pourquoi, dans la suite nous exploiterons aussi le taux d'accroissement (TA) caractérisant les motifs émergents – ce qui nous assurera une corrélation positive (voir la proposition 1 et sa preuve en appendice).

Proposition 1 *Soit un entier $\rho > 1$, seuil de valeurs minimum pour TA, et $\pi : X \rightarrow c_i$, une règle d'association concluant sur la classe c_i. Alors,*

$$TA(X, r_{c_i}) \geq 1 \Rightarrow X \text{ est positivement corrélé avec } c_i$$

Dans [HC06], les auteurs placent plusieurs mesures d'intérêt (dont $conf$ et TA) dans un cadre de travail plus général appelé les mesures δ-dépendantes. De telles mesures, notées m, dépendent de la fréquence γ du corps de la règle et du nombre d'erreurs que commet la règle dans r selon les deux principes suivants :

- *(i)* lorsque γ est fixe, $m(X, r)$ augmente avec $freq(\pi, r)$
- *(ii)* lorsque δ est fixe, $m(X, r)$ augmente avec γ.

Ce cadre de travail nous permet de dériver des bornes inférieures pour les mesures d'intérêt dites δ-dépendantes en fonction des valeurs de γ et δ. Vérifions-le pour la confiance et le taux d'accroissement. Considérons la table de contingence pour une règle δ-forte $X \rightarrow c_i$ en table 3.6.

7. δ-SCR pour δ-strong characterization rule
8. Ici fréquente veut dire $(\gamma - \delta)$-fréquente puisque γ est la fréquence du corps de π.

$\pi : X \rightarrow c_i$	c_i	$\bar{c_i}$	Σ
X	$\gamma - \delta$	δ	γ
$\bar{X}$	.	.	.
Σ	$\|r_{c_i}\|$	$\|r \setminus r_{c_i}\|$	$\|r\|$

FIGURE 3.6 – Table de contingence pour la règle δ-forte $\pi : X \rightarrow c_i$ et bornes en fonction de γ et δ.

Par construction, $(\gamma - \delta)$ est une borne inférieure pour $freq(X, r_{c_i})$. De même, δ est une borne supérieure pour $freq(X, r \setminus r_{c_i})$. Notons que par déduction nous pouvons obtenir d'autres bornes pour les autre cellules. Par conséquent, nous obtenons des bornes inférieures pour nos mesures TA et $conf$:

$$TA(\pi, r_{c_i}) \geq \frac{\gamma - \delta}{\delta} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}|} \quad \text{and} \quad conf(\pi, r) \geq 1 - \delta/\gamma$$

Nous utilisons ces bornes pour étendre la définition des δ-SCRs. Par la suite nous utiliserons cette définition.

Définition 16 (Règle δ-forte de caractérisation) *Soit $\gamma > 0$ un entier, seuil de fréquence minimum et $\delta > 0$ un entier, seuil d'erreurs maximum. $\pi : X \rightarrow c_i$ est une règle δ-forte de caractérisation (δ-SCR) si c_i est un attribut classe et le corps X est minimal. X est dit minimal s'il n'existe pas d'autre règle fréquente $\pi' : Y \rightarrow c_i$ telle que $Y \subseteq X$ et $conf(\pi', r) \geq 1 - \frac{\delta}{\gamma}$ et $TA(\pi, r_{c_i}) \geq \frac{\gamma-\delta}{\delta} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}|}$.*

Cette formalisation des δ-SCRs nous permet d'assurer qu'étant donné un seuil de taux d'accroissement $\rho > 1$, sous de simples conditions sur les valeurs de γ et δ, les corps de δ-SCRs de $S_{\gamma,\delta}$ sont des ρ-EPs. Nous traduisons ces conditions suffisantes dans la proposition 2 (voir sa preuve en appendice).

Proposition 2 *Soit $\gamma > 0$ et $\delta > 0$ deux entiers respectivement seuil de fréquence minimum et seuil de d'erreurs maximum. Soit $\rho > 1$ un seuil de taux d'accroissement et $\pi : X \rightarrow c_i$ une règle d'association δ-forte qui conclut sur une classe c_i. Alors,*

$$\delta < \frac{\gamma}{\rho} \cdot \frac{|r \setminus r_{c_j}|}{|r|} \quad \Longrightarrow \quad TA(X, r_{c_i}) \geq \rho, \textit{ ainsi } X \textit{ est un } \rho\textit{-EP} \tag{3.1}$$

$$\delta < \gamma/2 \quad \Longrightarrow \quad conf(\pi, r) \geq 1/2 \tag{3.2}$$

où c_j est la classe majoritaire dans r.

Ainsi, pour γ fixé, les itemsets γ-fréquents δ-libres X dont la δ-fermeture contiennent un attribut classe c_i et tels que δ satisfait les équations 3.1 et 3.2 de la proposition 2 forment un ensemble de ρ-EPs sans conflits. Dans la suite, γ et δ respecteront les contraintes des équations 3.1 et 3.2.

Extraction des règles δ-fortes de caractérisation

Pour extraire $S_{\gamma,\delta}$ l'ensemble des règles δ-fortes de caractérisation, nous utilisons une extension de l'algorithme par niveau d'extraction d'itemsets γ-fréquent δ-libres rappelé dans le chapitre 2. Aux contraintes anti-monotones de fréquence et de δ-liberté, notre extension rajoute les contraintes suivantes :

1. $C_1 \equiv \exists c \in \mathcal{C} \mid c \in cl_\delta(X, r)$ (contrainte syntaxique)
2. $C_2 \equiv \nexists Y \in S_{\gamma,\delta} \mid Y \subseteq X$ (contrainte de minimalité)

Ces deux contraintes ne font que réduire le nombre de motifs extraits et le temps d'extraction par rapport à l'implémentation `AC-like` qui extrait tous les itemsets γ-fréquents δ-libres X et leur δ-fermeture $cl_\delta(X, r)$. En effet, ici nous ne gardons que les itemsets X tels qu'il existe une classe c dans $cl_\delta(X, r)$, et si X respecte cette contrainte, ces sur-ensembles ne seront pas candidats car nous voulons les minimaux selon l'inclusion ensembliste. Dans la suite, nous proposons une méthode pour utiliser $S_{\gamma,\delta}$ pour construire des descripteurs robustes.

3.4 Vers de nouveaux descripteurs numériques

3.4.1 Nouveau codage numérique des descripteurs

L'idée clé est de construire un nouveau descripteur pour chaque règle δ-forte de caractérisation (δ-SCR) extraite. Le nouveau descripteur sera alors un nouvel attribut. Dans [CYHH07], les auteurs utilisent un codage binaire. C'est-à-dire, pour un objet t, la valeur du nouveau descripteur est 1 si la δ-SCR correspondante couvre t et 0 sinon. Dans cette section, nous proposons un codage numérique plus prometteur pour construire de nouveaux descripteurs. Le processus de construction de descripteurs `FC` est résumé par l'algorithme 8.

Tout d'abord, (ligne 2), la procédure `FeaturesExtraction` extrait tous les itemsets γ-fréquents δ-libres I' qui sont les corps de δ-SCRs dans r. Puis, chaque I' devient un nouvel attribut pour r'. Et, (ligne 5), la valeur de I' pour un objet t est la proportion d'attributs de I' qui sont vérifiés par t dans r. L'opérateur $Items$ est l'opérateur dual pour $Objets$. Ainsi, $\mathcal{R}' : (\mathcal{T} \times \mathcal{I}') \mapsto [0; 1]$ et $\mathcal{R}'(t, I')$ prend ses valeurs entre 0 et 1, plus

Algorithme 8 : `FC` : construction de descripteurs basés sur les motifs

input : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ une base de onnées binaires,
γ, entier positif, seuil de fréquence minimum
δ, entier positif, seuil d'erreurs maximum
output : $r' = \{\mathcal{T}, \mathcal{I}', \mathcal{R}'\}$, une nouvelle base de données numériques

1 **begin**
2 $\quad \mathcal{I}' \longleftarrow$ `FeaturesExtraction`(r, γ, δ);
3 $\quad$ **for** $t \in \mathcal{T}$ **do**
4 $\quad\quad$ **for** $I' \in \mathcal{I}'$ **do**
5 $\quad\quad\quad \mathcal{R}'(t, I') \longleftarrow \frac{|I' \cap \text{Items}(t,r)|}{|I'|}$;
6 $\quad r' \longleftarrow \{\mathcal{T}, \mathcal{I}', \mathcal{R}'\}$;
7 **end**

précisément :

$$\mathcal{R}'(t, I') \in \{0, \frac{1}{|I'|}, \ldots, \frac{|I' - 1|}{|I'|}, 1\}$$

Nous pensons que ce codage numérique est moins strict et plus pertinent qu'un simple codage binaire. En effet, une étape supplémentaire de discrétisation supervisée de tels descripteurs peut nous permettre d'obtenir une segmentation plus pertinente : dans le pire des cas, la segmentation du domaine de définition de I' se fera entre les valeurs $(|I'| - 1)/|I'|$ et 1 (ou 0 et $1/|I'|$) ; dans les autres cas, plusieurs segmentations plus prometteuses pourront se faire entre $(j-1)/|I'|$ et $j/|I'|$ où $1 \leq j \leq |I'|-1$. Enfin, (ligne 6), r' est notre nouvelle base de données construite à partir de nos nouveaux descripteurs. Notre processus de construction de descripteurs peut être résumé par le schéma de la figure 3.7.

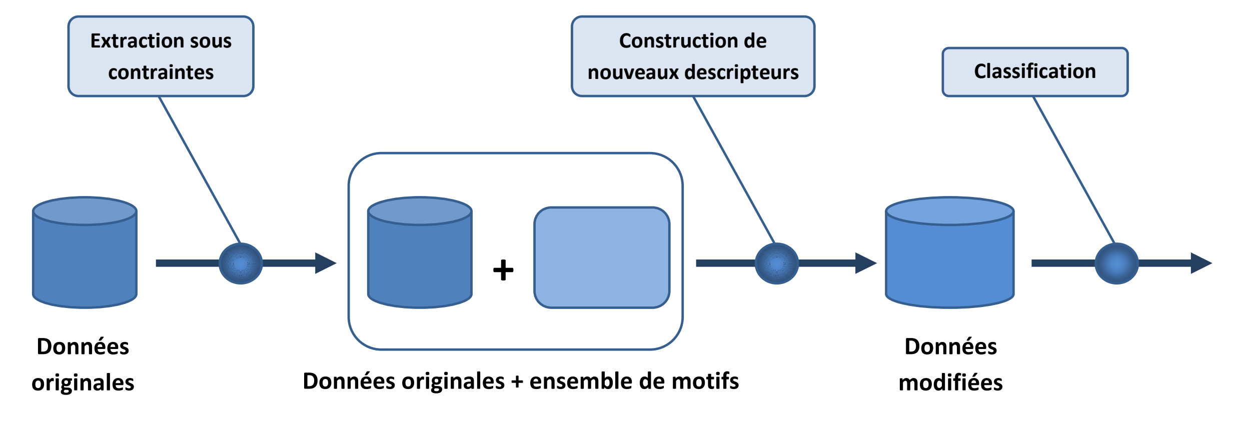


FIGURE 3.7 – Processus générique de construction de descripteurs à base de motifs

Dans la suite, nous expérimentons ce processus à travers diverses instantiations pour

le valider expérimentalement.

3.4.2 Paramétrage et validation dans les contextes bruités

Nous notons notre processus de construction de descripteurs FC. Notre validation expérimentale répondra aux questions suivantes :

- Q_1 Considérons des classifieurs dits classiques [WKQ+08] – l'arbre de décision C4.5 [Qui93], le classifieur naïf de Bayes NB [JL95] et les machines à vecteurs support (SVM). Ces classifieurs sont-il plus performants en terme de précision lorsqu'on utilise les nouveaux descripteurs générés par FC que lorsqu'on utilise les attributs originaux ?
- Q_2 Que se passe-t-il lorsque les attributs sont soumis au bruit ? FC procure-t-il de meilleurs résultats de précision que les classifieurs classiques ? En d'autres termes, FC est-il un processus tolérant aux bruits ? Comment choisir les valeurs de δ lorsque les données sont bruitées ?
- Q_3 Est-ce qu'un classifieur classique en fin de processus FC est comparable en terme de précision à un classifieur *avancé* basé sur les motifs ? – nous prendrons ici comme référence le classifieur HARMONY [WK05, WK06].

Protocole

Pour répondre aux trois questions précédentes, nous posons deux protocoles d'expérimentations : pour les données originales (non-bruitées) et pour les données artificiellement bruitées.

Données originales : Pour asseoir l'efficacité de notre processus, nous utilisons FC sur plusieurs bases de données UCI (voir une description des données en appendice). Pour mettre sur un point d'égalité les classifieurs classiques et les classifieurs classiques "branchés" à FC , lorsque les bases de données contiennent des attributs continus, nous procédons à une discrétisation supervisée [FI93], puis à une binarisation de la base d'apprentissage. Ce schéma est ensuite reporté sur les données test.

Données aux attributs bruités : Nous voulons aussi étudier le comportement de FC en présence de bruits dans les données. Dans un tel contexte, nous voulons être capables d'apprendre des modèles prédictifs performants malgré la présence de bruit dans les attributs. Pour évaluer la résistance de FC aux bruits d'attributs, dans nos expériences, nous traiterons des données d'apprentissage artificiellement bruitées et des données test *propres* (non-artificiellement bruitées). Pour ce faire, nous ajoutons du bruit aléatoirement

à différentes quantités seulement dans les données d'apprentissage de la manière suivante : Pour une base de données r et un niveau de bruit de $x\%$, ($x \in \{10, 20, 30, 40, 50\}$), chaque attribut $a \in \mathcal{I}$ a $x\%$ de chances de prendre une autre valeur de son domaine de définition (incluant sa valeur courante) dans chaque objet $t \in \mathcal{T}$ de l'ensemble d'apprentissage. Lorsque les attributs originaux sont continus, ils sont tout d'abord discrétisés comme auparavant, puis on y injecte du bruit, enfin on procède à la binarisation – ce qui nous évite qu'un objet soit décrit par plusieurs intervalles de valeurs d'un même attribut anciennement numérique.

Dans nos expériences, toutes les étapes de pré-traitement (injection de bruit, discrétisation, binarisation), ainsi que les résultats de précision pour `C4.5` , `NB` , `SVM` , `FC-C4.5` , `FC-NB` et `FC-SVM` sont obtenus par 10-CV en utilisant la plateforme `WEKA` [WF05] et avec les bibliothèques `LibSVM` et `WLSVM` [CL01, EM05]. Tous les résultats de précision, ainsi que ceux de `HARMONY` sont obtenus sur la même 10-CV pour une comparaison plus pertinente. Nous utilisons le prototype d'`HARMONY` fourni par ses auteurs pour obtenir les résultats de précision.

Paramétrage

Notre processus dépend donc fortement des deux paramètres : le seuil de fréquence minimum γ et le seuil d'erreurs maximum admises δ. Nous avons vu que les valeurs extrêmes pour γ ne sont pas la solution : pour les plus basses valeurs de γ, l'ensemble des règles $s_{\gamma,\delta}$ peut être un très grand et parmi ces règles, beaucoup apporte très peu d'information puisqu'elles ne concernent qu'un petit ensemble d'objets ; d'un autre côté, pour les plus hautes valeurs de γ, $S_{\gamma,\delta}$ contient trop peu de règles pour couvrir convenablement l'ensemble d'apprentissage. En vérité, sélectionner un seuil judicieux de fréquence minimum γ est toujours un problème ouvert. Il existe des solutions préliminaires à ce problème (e.g.[ZWZL08]). Et dans le chapitre 4, nous proposons une méthode pour sélectionner automatiquement des seuils de fréquence minimum. Toutefois pour le chapitre présent, nous nous focaliserons sur la sélection du paramètre δ étant donné γ.

Étant donné un seuil de fréquence minimum γ, comment déterminer une valeur de δ pertinente ? L'évolution des mesures d'intérêt δ-dépendantes (telles que TA et $conf$) sont bien connues. Des valeurs décroissantes de δ impliquent des valeurs croissantes pour TA mais de tels motifs peuvent être rares et/ou non-pertinents dans des données bruitées. D'autre part, lorsque δ augmente, à l'instar de la caractérisation de groupes par des δ-bi-ensembles, les motifs extraits pourront représenter des motifs bruités dans les données d'apprentissage ; mais de plus grandes valeurs de γ détériorent la qualité des motifs de $S_{\gamma,\delta}$ (i.e. TA décroît lorsque δ augmente puisqu'on autorise plus d'erreurs). Dans la suite nous proposons une stratégie pour déterminer δ. Nous motivons notre stratégie par une étude expérimentale de l'évolution des résultats de précision en fonction des valeurs δ.

Une simple stratégie pour déterminer δ : En figure 3.8, 3.9 et 3.10, nous représentons les résultats de précision en fonction de des valeurs de δ pour diverses valeurs de γ et de niveau de bruit, pour `FC-C4.5` sur les données `tic-tac-toe` (figure 3.8), pour `FC-NB` sur les données `horse-colic` (figure 3.9) et pour `FC-SVM` sur les données `heart-cleveland` (figure 3.10). Pour référence, nous reportons aussi la précision obtenue avec le classifieur classique en question dans les mêmes conditions de bruits (voir la droite en gras dans chaque graphe). Nous remarquons que la précision de `FC` augmente (pas nécessairement de manière monotonique) avec δ jusqu'à un point maximal (pour δ_{max})– souvent meilleur que la précision obtenue avec le classifieur classique sur les données aux attributs originaux. Ensuite, la précision de `FC` décroît. Nous avons représenté ici des graphiques pour trois bases et trois classifieurs classiques mais ce comportement est le comportement général de la précision `FC` vis-à-vis de δ que nous pouvons observer aussi sur d'autres bases de données. Notons aussi que plus il y a de bruit d'attribut, plus δ_{max} est grand. De même, si l'on considère δ_{sup} les plus petites valeurs de δ pour lesquelles `FC` produit une meilleure précision que le classifieur classique, on observe que plus il y a de bruit, plus δ_{sup} doit être grand pour "capter" le bruit et obtenir de meilleures performances que le classifieur classique – ce qui confirme bien les valeurs de δ sont liées à la quantité de bruit dans les données.

En figure 3.11, nous représentons les résultats de précision en fonction du niveau de bruit d'attribut dans les données pour différentes valeurs de γ (une par graphe) et différentes valeurs de δ (une par courbe). Nous remarquons que pour de faibles niveaux de bruit, de petites valeurs de δ suffisent pour assurer de meilleurs résultats de précision pour `FC-C4.5` comparé à `C4.5` . Lorsque le bruit augmente, `FC-C4.5` avec des petites valeurs de γ – i.e. qui utilise des règles *presque* fortes – obtient de piètres résultats de précision par rapport à `C4.5` . Ainsi, essayer de trouver des corrélations fortes dans des données bruitées se révèle inefficace. A l'inverse, pour des valeurs de δ plus grandes, `FC-C4.5` semble mieux capter des motifs bruités qu'avec de petites valeurs. Notons aussi que lorsque `FC-C4.5` obtient de meilleurs résultats de précision, ce n'est pas le fait d'un seuil γ.

Maintenant que nous avons une meilleure compréhension des relations entre les paramètres γ, δ et le niveau de bruit, ainsi que de leur impact sur les résultats de précision de `FC` , nous proposons une stratégie pour déterminer automatiquement une valeur pertinente pour δ. Comme nous ne connaissons pas a priori la quantité de bruit dans les données et que nous ne pouvons pas utiliser la précision sur les données test, notre stratégie s'adaptera à l'importance du bruit via les données d'apprentissage. En figure 3.12, nous représentons la précision sur les données d'entraînement de `FC-C4.5` en fonction de δ pour diverses valeurs de γ et divers niveaux de bruit pour `tic-tac-toe`. Comme attendu, la précision d'entraînement augmente avec δ jusqu'à stabilisation (pour des valeurs notées δ_{opt}). Ces δ_{opt} sont particulièrement intéressantes car pour $\delta < \delta_{opt}$ la précision d'entraînement est moindre qu'avec δ_{opt}, et pour $\delta > \delta_{opt}$, l'amélioration de la précision d'entraînement par rapport à $\delta < \delta_{opt}$ n'est pas significative voire nulle – car les règles extraites sont moins

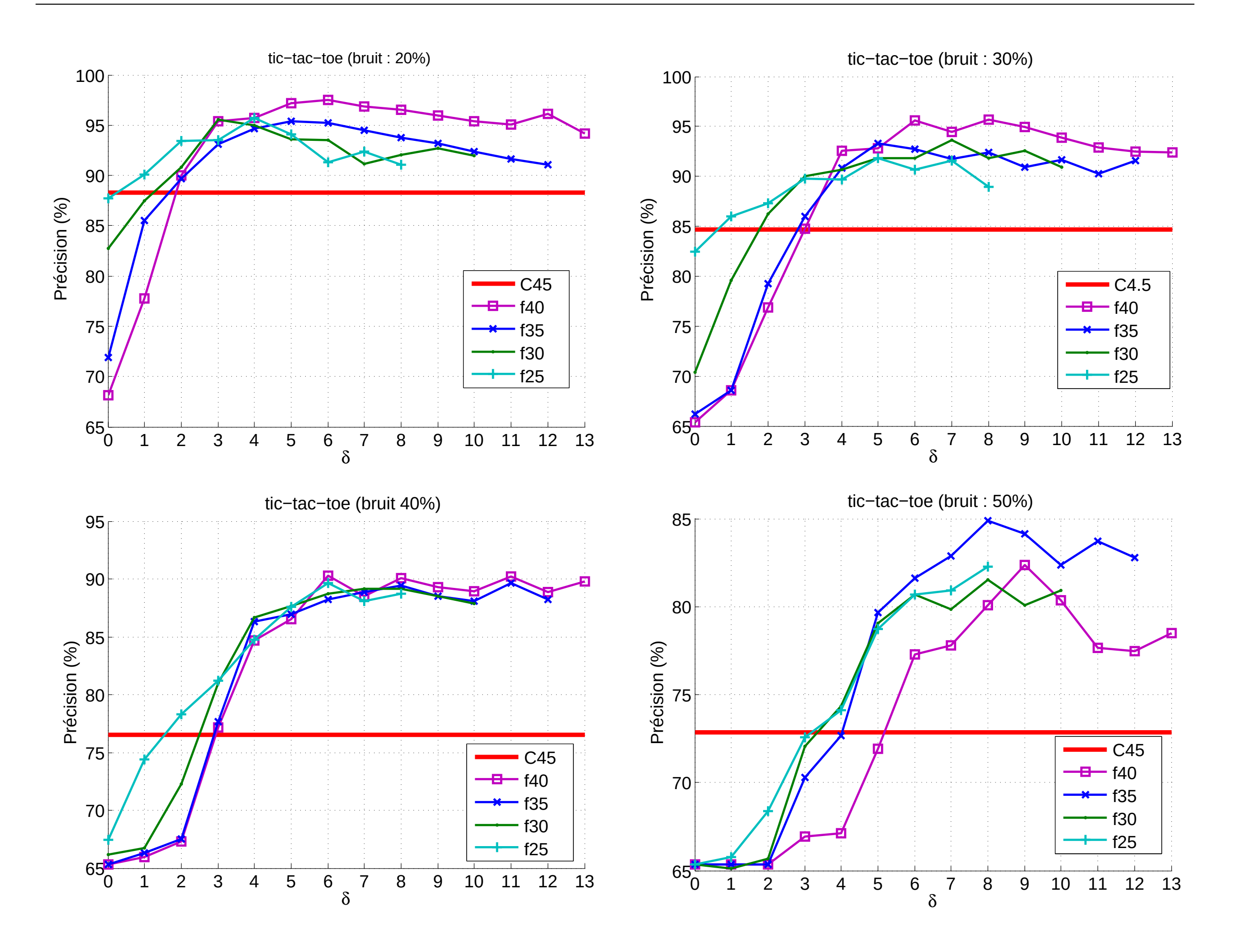


FIGURE 3.8 – Evolution de la précision de FC-C4.5 en fonction de δ pour divers niveaux de bruits et seuils de fréquence pour les données tic-tac-toe.

pertinentes. Comme le niveau de bruit n'est pas connu a priori, nous proposons la stratégie suivante pour déterminer une valeur proche de δ_{opt} :

1. Augmenter δ en partant de 0,
2. Analyser la précision d'entraînement jusqu'à stabilisation.

Notons que par souci d'adaptatilité, nous analysons ici la précision d'entraînement plutôt que la couverture de base d'entraînement pour déterminer les δ_{opt}. En effet, considérons $S_{\gamma,\delta}$ l'ensemble des motifs extraits ; le taux de couverture de la base d'apprentissage (et donc δ_{opt} choisi stratégiquement) est le même quel que soit le classifieur classique en fin de FC . Malheureusement, nous observons expérimentalement que les "bonnes" valeurs de δ ne sont pas forcément les mêmes selon le classifieur branché en fin de processus. Ceci est tout simplement dû au fonctionnement même des différents classifieurs. C'est pourquoi nous préférons analyser l'évolution de la précision d'entraînement afin d'obtenir différentes valeurs de δ_{opt} selon le classifieur en fin de processus.

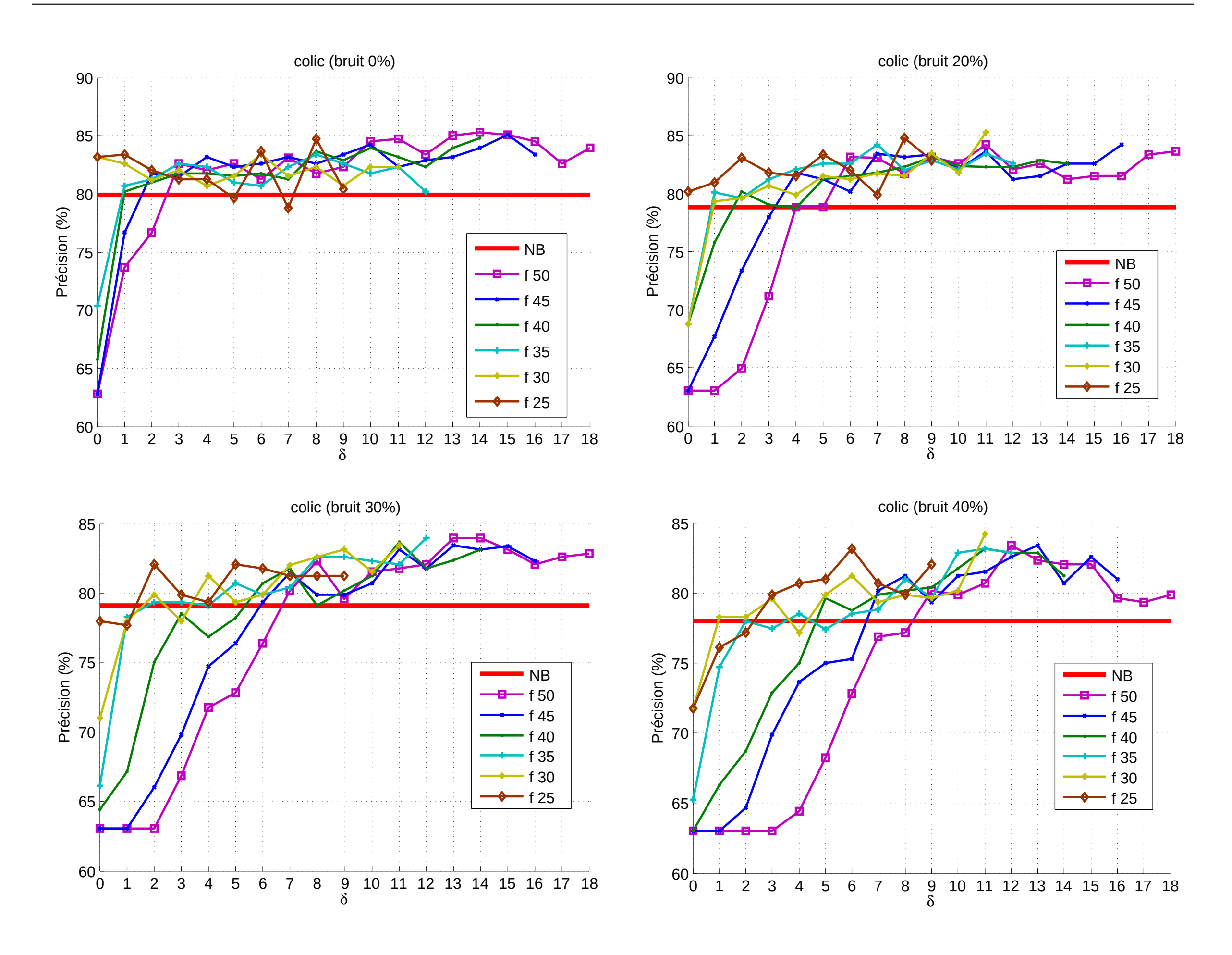


FIGURE 3.9 – Evolution de la précision de FC-NB en fonction de δ pour divers niveaux de bruits et seuils de fréquence pour les données colic.

Résultats de précision et comparaison

Tous les résultats de précision obtenus par 10-CV sont reportés dans la table 3.5. Nous ferons deux types de comparaisons de précision : premièrement nous comparons les résultats de précision des classifieurs classiques (C4.5 , NB , SVM-RBF [9]) sur les données aux attributs originaux (éventuellement bruités) avec les précisions du même classifieur classique branché en fin de FC . Deuxièmement, nous comparons le processus FC avec HARMONY . Pour chaque base, nous utilisons divers seuils de fréquence minimum γ, et pour chaque valeur de γ, nous appliquons notre stratégie énoncée précédemment pour déterminer δ. Ainsi, les résultats de FC sont reportés en deux colonnes dans la table 3.5 : *(i)* la colonne FC reporte la moyenne des résultats de précision pour toutes les valeurs de

9. RBF pour Radial Basis Function.

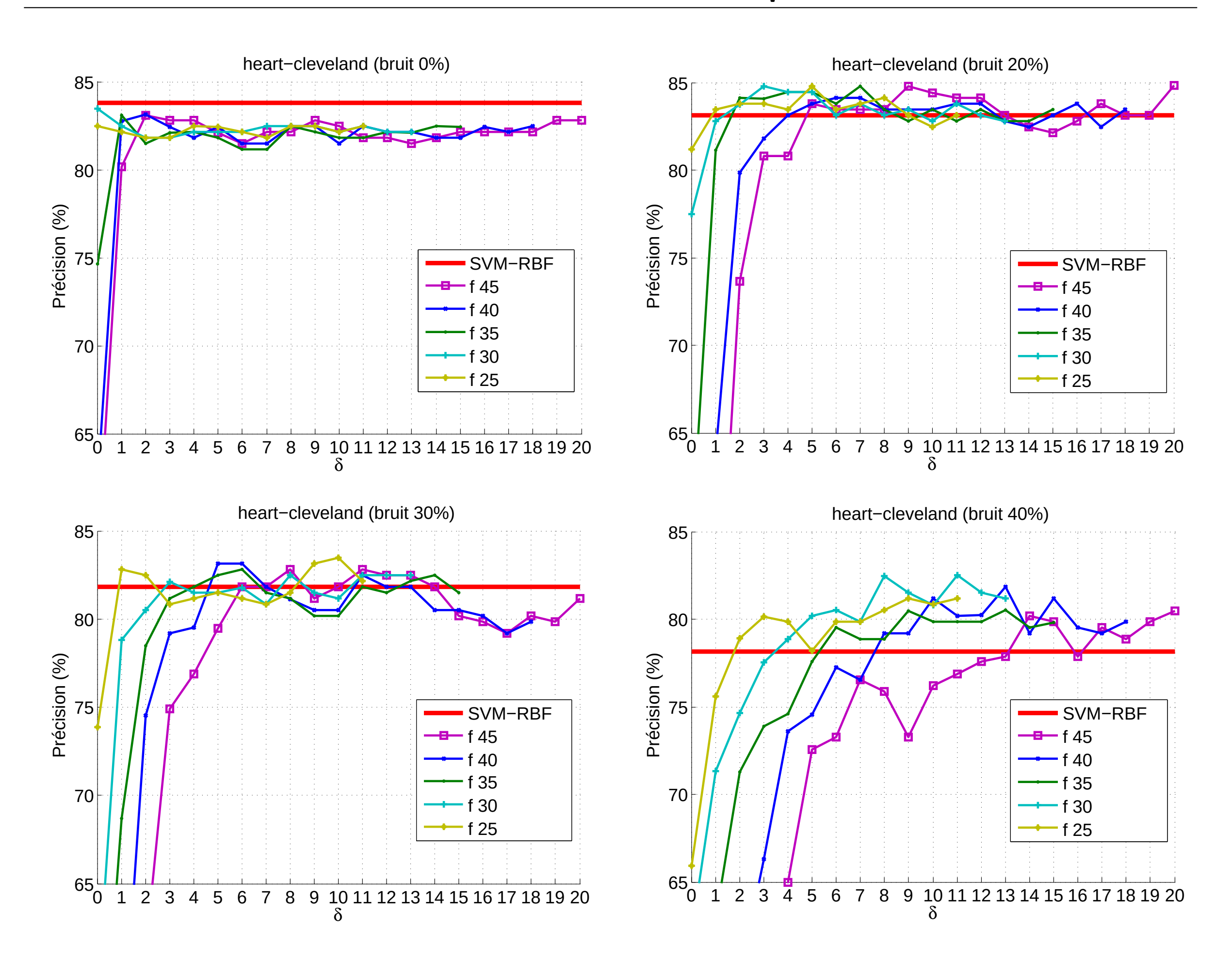


FIGURE 3.10 – Evolution de la précision de FC-SVM en fonction de δ pour divers niveaux de bruits et seuils de fréquence pour les données `heart-cleveland`.

γ testées et en utilisant δ_{opt} ; *(ii)* la colonne *Max* reporte la meilleure précision obtenue pour un certain γ en utilisant le $\delta_{(opt)}$ correspondant.

Tout d'abord nous remarquons que sur les données non-bruitées, `FC-C4.5` et `FC-NB` obtiennent de meilleures précisions que les classifieurs classiques respectifs (`FC-C4.5` gagne 10 fois sur 11 contre `C4.5` et `FC-NB` 7/11 contre `NB`). Si nous considérons une bonne valeur de γ (voir colonne *Max*), les résultats de comparaison sont de 11/11 pour `FC-C4.5` et de 8/11 pour `FC-NB` . En ce qui concerne `SVM-RBF` , les résultats sont moins significatifs : 4/11 et 7/11 pour une bonne valeur de γ. Ici, `SVM-RBF` branché à `FC` est utilisé avec son paramétrage par défaut ; nous pensons que ce non-paramétrage est la cause des pauvres résultats de précision pour `FC-SVM` .

Lorsque les données sont bruitées, `FC` est à l'oeuvre et les classifieurs appris sur les données améliorées par les nouveaux descripteurs robustes sont plus performants. `FC-C4.5`

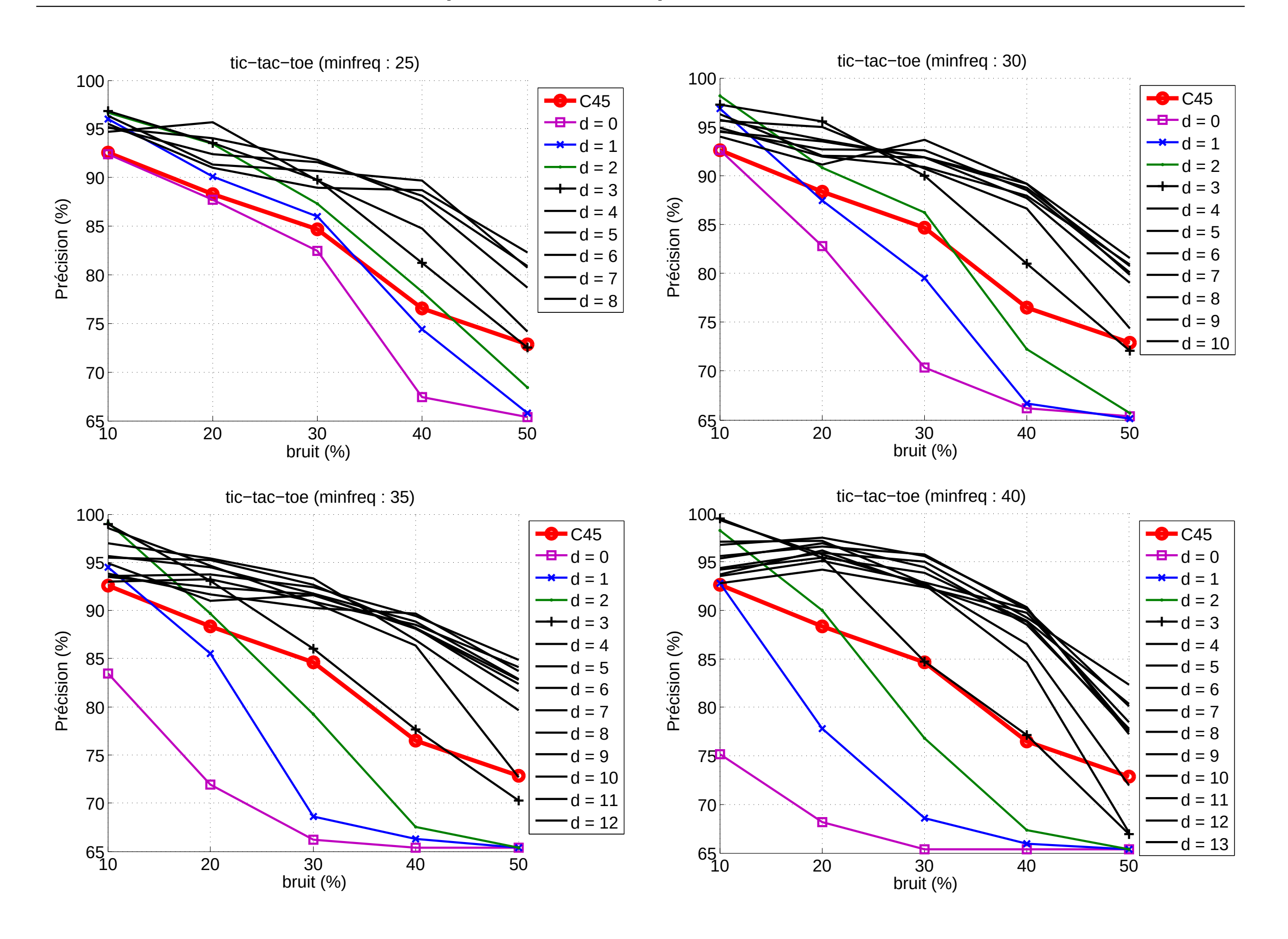


Figure 3.11 – Evolution de la précision de FC-C4.5 en fonction du bruit pour différents seuils de γ et δ pour les données tic-tac-toe.

gagne 35 fois sur 55 contre C4.5 et pour FC-NB et FC-SVM les ratios sont respectivement 28/55 et 40/55 (voir les résultats en gras). Une fois encore, si nous considérons une bonne valeur pour γ nous avons de meilleurs ratios : respectivement 50/55, 41/55 et 42/55. Dans la table 3.5, nous reportons aussi l'amélioration moyenne de la précision sur toutes les données qu'apporte le processus FC . Nous voyons clairement que pour des données bruitées, dans la plupart des cas, l'apport de la nouvelle description des données offerte par FC nous permet d'obtenir de meilleurs résultats de précision. Toutefois, pour FC-NB , les résultats doivent être contrastés. Les résultats montrent que NB est naturellement plus résistant au bruit que les autres classifieurs classiques, donc les résultats de FC-NB sont moins impressionnants. Pour un haut niveau de bruit dans les données, les résultats en moyenne sont même moins bons, surtout lorsque nous ne disposons pas d'une bonne valeur de γ.

Nous avons reproduit des résultats de classification pour HARMONY en utilisant la même validation croisée pour une comparaison plus judicieuse. Comme conseillé par les auteurs

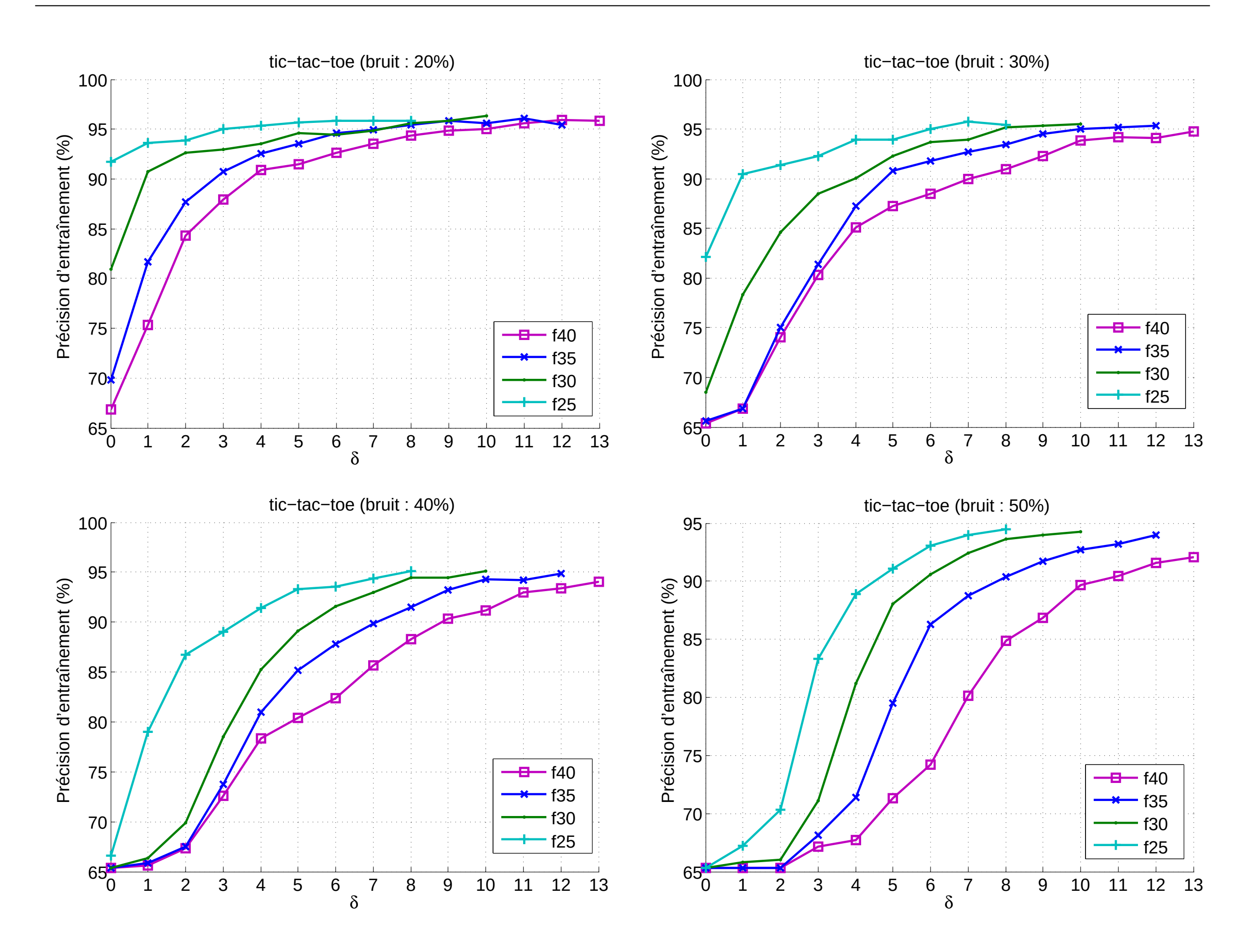


FIGURE 3.12 – Evolution de la précision d'entraînement de FC-C4.5 en fonction de δ pour différents seuils de γ et de bruit pour les données tic-tac-toe.

dans [WK05, WK06], nous utilisons HARMONY avec différents seuils de fréquence (5,10,15) et reportons le meilleur résultat de précision. Nous comparons donc HARMONY avec les résultats de FC en colonnes *Max*. Les classifieurs classiques (C4.5 , NB , SVM-RBF) gagnent respectivement 9/66, 37/66 et 33/66 contre HARMONY . Excepté pour NB , HARMONY est généralement plus performant que C4.5 et comparable à SVM-RBF . Pour avoir une idée de l'amélioration de la précision qu'apporte FC , nous comptons combien de fois un classifieur classique perd face à HARMONY et gagne lorsqu'il est en fin de processus FC . FC-C4.5 affiche un résultat de 23, FC-NB 12 et FC-SVM 9. Une fois encore il est intéressant d'utiliser FC pour améliorer la description des données et par conséquent les résultats de précision des classifieurs appris. En effet, avec la nouvelle description générée par FC , des classifieurs classiques comme C4.5 , NB , SVM-RBF deviennent comparables voire meilleurs que HARMONY . FC-C4.5 gagne 32 fois sur 66, FC-NB 44/66 et FC-SVM 40/66.

3.5 Discussion et limites

Si nous revenons aux trois points-clés de la classification associative énoncés en introduction : intérêt des règles, couverture des données d'apprentissage, concision et non-redondance de l'ensemble de règles ; nous voyons que FC propose une solution de choix à chacun d'eux. Les contraintes sur les valeurs de γ et δ nous assurent des règles de bonne qualité. Notre stratégie pour ajuster δ en fonction de γ nous assure d'obtenir une bonne couverture positive des données d'apprentissage. Enfin, le fait d'utiliser l'ensemble des règles δ-fortes de caractérisation, nous évitons des redondances de règles et obtenons un ensemble concis. De plus, nous profitons de la robustesse déjà prouvée dans d'autres tâches de fouille pour construire des nouveaux descripteurs numériques résistants aux bruits d'attributs.

Toutefois, on peut identifier au moins un type de problèmes pour lequel FC peut faillir : les problèmes où les classes sont multiples et disproportionnées (en nombre d'objets). Dans ces cas-là, comme nous utilisons un (seul) seuil de fréquence global, pour espérer caractériser les objets de la classe minoritaire, une valeur de γ (très) basse est nécessaire. La valeur de δ étant contrainte par celle de γ doit être plus basse (voire proche de zéro) – on est alors prêt à rechercher des règles exactes. Si les données sont quelque peu bruitées FC sera alors moins performante en raison du faible intérêt des règles extraites. Il faut noter que dans ce cas, l'extension des techniques de classification associative à plusieurs seuils de fréquence (un par classe) est possible (e.g. HARMONY). Toutefois, elle est loin d'être triviale dans notre cadre de travail. En effet, les valeurs de γ et de δ étant liées, étendre notre cadre de travail à plusieurs seuils de fréquence demanderait aussi plusieurs seuils d'erreurs et donc de plus amples investigations.

Dans le chapitre suivant, nous nous intéressons plus particulièrement à ces problèmes d'actualité en classification supervisée : les problèmes multi-classes disproportionnées.

FC et bruit de classe

En parallèle de notre travail présenté dans ce mémoire, le processus FC a été aussi éprouvé dans des données où les attributs classes sont bruités. L'application directe de FC (originellement dédié aux contextes dont les attributs non-classe sont bruités) à ce type de problèmes fournit des résultats décevants. En effet, les classifieurs classiques résistent mieux au bruit de classe sur les données avec les descripteurs originaux qu'avec les nouveaux descripteurs fournis par FC . Nous pensons que cette divergence de résultats – entre données aux attributs bruités et données aux classes bruitées – vient de la différence des impacts des différents types de bruits. En effet, la présence de bruit dans les attributs d'un objet t de classe c_i a peu de chances de transformer cet objet en quelque chose de similaire à un objet de classe c_j. Au contraire, le bruit de classe, par définition, change la classe

d'un objet. La confusion ainsi générée est donc plus grande. Considérons un ensemble d'objets T de classe c_i ayant certaines caractéristiques I en commun et qui ont quasiment tous changés de classe vers c_j. `FC` pourrait considérer I comme caractéristique de la classe c_j tout en occultant d'autres caractéristiques que T ou un de ces sous-ensembles aurait avec d'autres objets de c_i. Ce qui impliquera des erreurs de classification dans la phase test.

Data sets	C4.5	FC-C4.5	Max	NB	FC-NB	Max	SVM	FC-SVM	Max	HARMONY
breast-w	95.57	**95.62**	**95.85**	97.28	*96.59*	*96.85*	96.85	**97.19**	**97.42**	95.85
10%	95.14	*94.59*	**95.14**	96.99	**97.22**	**97.56**	96.56	**97.19**	**97.42**	95.71
20%	93.28	*92.31*	*92.99*	96.99	**97.05**	**97.14**	95.85	*95.82*	**96.14**	95.71
30%	89.56	**91.00**	**91.85**	96.85	**97.14**	**97.28**	95.14	**95.16**	**95.42**	94.99
40%	87.70	**87.73**	**88.41**	96.42	*94.99*	**97.28**	92.56	*91.28*	*91.85*	93.56
50%	87.56	*86.30*	**87.56**	96.28	**96.51**	**97.21**	86.41	*83.06*	*83.98*	90.99
colic	85.04	*84.31*	**85.85**	79.90	**82.73**	**85.31**	84.77	*84.59*	**84.77**	82.88
10%	83.14	*82.95*	**85.29**	78.81	**82.59**	**84.77**	84.50	**85.17**	**86.12**	82.06
20%	82.04	*80.88*	**83.12**	78.81	**81.28**	**85.31**	83.94	**84.88**	**85.82**	81.79
30%	80.95	*80.78*	**82.85**	79.09	**81.37**	**83.96**	80.98	**84.61**	**85.28**	82.88
40%	80.93	**81.72**	**84.21**	78.00	**82.32**	**84.22**	69.55	**81.87**	**85.30**	83.15
50%	81.53	**82.04**	**83.66**	73.92	**80.56**	**82.31**	63.05	**76.14**	**81.51**	82.61
diabetes	73.58	**73.96**	**74.23**	73.96	*73.61*	**75.00**	74.74	**75.05**	**75.56**	73.05
10%	72.79	**72.79**	**73.31**	73.31	*72.69*	**73.83**	73.44	**75.19**	**75.39**	73.70
20%	72.02	**72.02**	**72.27**	73.83	*72.50*	*73.32*	72.92	*71.58*	*72.61*	73.70
30%	71.23	**72.02**	**73.06**	73.44	*72.56*	**73.70**	72.01	*65.24*	*65.62*	72.66
40%	68.88	**69.57**	**70.96**	75.13	*71.76*	*72.97*	70.06	*65.11*	*65.11*	70.96
50%	68.37	*67.94*	**69.15**	70.45	*67.55*	*68.75*	67.59	*65.11*	*65.11*	68.10
heart-c	80.47	**82.48**	**84.13**	81.79	**83.20**	**84.12**	83.77	**83.88**	**84.10**	82.18
10%	78.83	**80.17**	**81.18**	82.78	**83.39**	**84.78**	83.12	**84.64**	**85.09**	81.19
20%	75.85	**79.46**	**82.18**	83.12	**83.26**	**84.46**	82.48	**84.59**	**84.79**	80.86
30%	77.19	**78.40**	**80.21**	84.45	*83.93*	**84.82**	81.82	**83.13**	**83.79**	82.84
40%	77.50	*76.40*	**78.52**	84.45	*82.63*	**84.48**	79.16	**81.90**	**82.84**	78.88
50%	72.28	**75.53**	**77.52**	79.52	**79.93**	**88.20**	77.48	**79.14**	**80.51**	76.24
heart-h	75.55	**79.60**	**81.64**	84.07	*82.50*	*83.03*	81.38	**82.49**	**83.38**	82.31
10%	77.62	**78.83**	**80.56**	83.73	*80.70*	*81.31*	82.02	**83.31**	**83.71**	84.01
20%	76.60	**78.24**	**80.29**	83.39	*82.30*	*83.00*	82.70	*81.88*	*82.67*	80.61
30%	75.56	**79.08**	**80.00**	84.05	*83.10*	**84.34**	81.32	**82.34**	**82.67**	79.93
40%	74.17	**79.19**	**80.96**	83.37	*80.51*	*82*	80.29	**82.40**	**83.01**	81.63
50%	73.83	**79.93**	**81.68**	79.95	**80.17**	**82.34**	68.63	**81.83**	**82.00**	78.57
heart-s	81.85	**83.33**	**86.67**	81.48	**82.10**	**84.45**	83.33	*83.18*	**83.70**	81.48
10%	80.74	**81.17**	**84.45**	81.48	*81.17*	**85.92**	83.70	*83.40*	**84.07**	81.85
20%	78.52	**79.75**	**80.74**	82.59	*81.60*	**84.07**	80.00	**83.27**	**84.45**	80.74
30%	74.07	**74.07**	**76.30**	81.11	**81.92**	**84.07**	78.15	**83.52**	**84.81**	77.41
40%	72.96	*70.15*	**73.33**	81.85	*71.70*	*72.59*	72.59	*58.81*	*63.70*	70.74
50%	59.26	**64.07**	**65.55**	55.55	**61.18**	**65.18**	59.63	*55.55*	*55.55*	63.70
hepatitis	81.87	**82.71**	**85.17**	82.00	**82.22**	**84.50**	83.21	*79.69*	*81.33*	83.87
10%	80.62	*80.19*	**81.33**	83.25	*82.90*	**84.50**	80.67	*79.37*	*79.37*	83.87
20%	78.62	**78.93**	**79.83**	85.21	*84.46*	**86.42**	79.37	*79.37*	*79.37*	81.29
30%	79.29	*78.23*	**81.08**	86.46	*83.81*	*84.37*	79.37	**79.37**	**79.37**	81.93
40%	76.04	**78.41**	**79.79**	85.79	*82.34*	*83.79*	79.37	**79.37**	**79.37**	78.06
50%	80.58	*79.46*	**83.79**	79.37	**79.81**	**81.21**	79.37	**79.37**	**79.37**	81.93
iris	93.33	**93.33**	**93.33**	92.67	**94.00**	**94.00**	94.67	*94.00*	*94.67*	95.33
10%	92.67	**93.00**	**94.67**	92.67	**92.67**	**94.00**	94.67	*94.00*	**94.67**	90.00
20%	90.00	*89.22*	**93.33**	92.67	**93.11**	**94.67**	94.00	**94.00**	**94.67**	90.67
30%	90.00	*86.50*	*89.33*	92.67	*91.33*	*92.00*	92.00	**92.00**	**94.67**	91.33
40%	86.00	*84.67*	*84.67*	92.67	*88.00*	*88.33*	90.67	*88.80*	**90.67**	90.00
50%	84.00	*77.33*	*77.33*	94.00	*77.33*	*77.33*	90.67	**80.77**	**84.00**	82.00
t-t-t	93.21	**99.40**	**100.00**	68.47	**79.72**	**81.73**	87.79	*86.40*	*89.24*	97.08
10%	92.06	**97.65**	**99.17**	68.57	**75.99**	**78.81**	81.10	*76.90*	*77.97*	96.97
20%	88.95	**96.26**	**97.18**	70.14	**73.81**	**74.42**	71.92	*65.34*	*65.45*	94.68
30%	82.78	**93.00**	**95.62**	71.92	**73.56**	**74.53**	65.65	*65.34*	*65.34*	91.13
40%	77.98	**89.43**	**91.55**	74.11	*73.56*	**74.83**	65.34	**65.34**	**65.34**	89.67
50%	71.39	**83.82**	**85.60**	70.57	**71.16**	**71.39**	65.34	**65.34**	**65.34**	81.84
wine	91.08	**91.53**	**93.76**	98.89	*92.92*	*93.86*	98.33	*96.91*	*97.22*	96.63
10%	91.63	**91.87**	**94.38**	96.6	*93.24*	*94.41*	98.30	*96.48*	*97.19*	95.50
20%	91.01	**91.74**	**92.71**	96.04	*94.42*	*95.00*	97.19	**97.19**	**97.19**	96.07
30%	84.37	**88.15**	**91.04**	94.38	*91.73*	**94.97**	97.19	**97.19**	**97.74**	92.70
40%	82.02	**88.28**	**88.82**	93.27	*91.42*	**93.79**	96.08	*95.24*	**96.63**	91.01
50%	82.12	**85.11**	**87.58**	88.10	**89.29**	**90.42**	91.60	*68.76*	*69.74*	88.76
vote	95.19	**96.32**	**96.78**	90.37	**92.14**	**92.20**	95.41	*95.29*	**95.41**	94.71
10%	94.26	**94.53**	**94.94**	89.68	**92.20**	**92.44**	95.64	*95.13*	*95.42*	95.63
20%	92.42	**93.78**	**94.24**	89.91	**91.18**	**91.59**	95.41	*94.84*	*95.18*	94.48
30%	93.11	*91.25*	*91.96*	88.98	**90.77**	**91.28**	94.72	*94.38*	*94.49*	94.94
40%	90.59	*90.56*	**91.72**	89.21	**90.71**	**91.74**	93.58	*92.91*	*93.11*	92.64
50%	89.44	*88.58*	**89.21**	89.21	**90.19**	**90.36**	92.20	*89.78*	*90.13*	93.10
Average		+1.44	+2.79		+0.99	+2.2		-0.51	+0.23	
10%		+0.75	+2.27		+0.63	+2.22		*-0.27*	+0.25	
20%		+1.21	+2.69		+0.21	+1.52		+4.74	+5.25	
30%		+1.31	+3.20		*-0.20*	+1.08		+0.36	+0.99	
40%		+1.94	+3.47		*-2.21*	-0.75		*-0.57*	+0.70	
50%		+1.80	+3.48		*-0.29*	+1.62		*-1.56*	-0.43	

Table 3.5 – Résultats de précision des classifieurs classiques, de FC et de HARMONY

Chapitre 4

Vers une solution pour les classes inégalement distribuées

Sommaire

4.1 Introduction et problématiques

4.1.1 Contexte général

Lorsque les différentes classes de données ne sont pas également fournies en nombre d'objets, on parle alors de problèmes aux classes inégalement distribuées ou de données "mal balancées". Souvent, dans les cas réels, les données à deux classes sont composées

d'une grande proportion d'objets dits normaux (de la classe majoritaire) et d'une petite proportion d'objets dits anormaux ou intéressants. Une autre manière de voir les choses est de considérer que le coût de mal classer un objet anormal comme objet normal est bien plus élevé que l'inverse.

Au vu des ateliers spécialisés [CJK04], le problème des classes disproportionnées a été un problème très étudié ces dernières années, et est toujours d'actualité [CJZ09]. Les approches les plus communes utilisent le principe d'échantillonnage : soit l'on considère un sous-échantillonnage des objets de la classe majoritaire, soit l'on considère un sur-échantillonnage de la classe minoritaire – les deux approches ayant pour but de rééquilibrer la distribution des classes. Si le sous-échantillonnage implique des pertes d'informations potentiellement importantes contenues dans les objets, le sur-échantillonnage en plus de rajouter des objets au risque de rendre la tâche plus laborieuse, peut produire des effets de sur-apprentissage.

Dernièrement, différents chercheurs se sont intéressés à des approches de classification supervisée basées sur des règles pour le problème des classes disproportionnées. Par exemple, dans [NH05], les auteurs proposent d'élaguer les corps de règles inductives issues d'un arbre de décision. Plus récemment, certains chercheurs ont pointé du doigt les défauts des approches de classification associative basées sur le cadre fréquence-confiance. Pour y remédier, [AC06, VC07] utilisent des règles de classification dont les corps sont positivement corrélés avec une classe.
Toutefois des problèmes persistent et dans la suite, nous montrons par l'exemple que lorsque nous sommes face à des problèmes à n classes (dits multi-classes lorsque $n > 2$), les différentes approches existantes basées sur le cadre fréquence-confiance, sur les motifs émergents ainsi que celles basées sur la corrélation positive montrent leurs limites. Nous montrons que les problèmes rencontrés par ces différentes approches sont dus au fait que la répartition des classes et/ou la fréquence des motifs extraits dans chacune des classes du problème. Nous montrons que ces problèmes sont inhérents au cadre de travail utilisé : le cadre `OVA` (`One-Versus-All`) dans lequel les motifs extraits sont caractéristiques d'une classe par rapport à l'union des autres classes. La solution que nous proposons pour résoudre ces problèmes identifiés nous amène à définir un nouveau cadre de travail de type `OVE` (`One-Versus-Each`) pour l'extraction de motifs intéressants (i.e. des règles de caractérisation `OVE`). Dans ce cadre, la répartition des classes et la répartition des motifs extraits est prise en compte. Pour ce faire, nous utilisons un seuil de fréquence minimum pour chaque classe et un seuil de fréquence maximum pour chaque type d'erreur de classification d'une classe c_i vers une classe c_j. Le nombre conséquent de paramètres seuils nécessitant une méthode automatique, nous proposons un algorithme d'optimisation `fitcare` [CGSB08] pour déterminer des paramètres prometteurs. Un algorithme d'extraction des règles de caractérisation `OVE` ainsi qu'un classifieur basé sur l'ensemble des règles `OVE` est aussi proposé et expérimenté sur des bases de données multi-classes disproportionnées.

4.1.2 Exemple motivant

A travers l'exemple simple suivant, nous montrons les limites rencontrées par les approches OVA.

Exemple 1 (Limites des approches OVA à base de motifs.) *Considérons la base de données r à trois classes ($\mathcal{C} = \{c_1, c_2, c_3\}$) décrite en figure 4.1. La répartition des objets de r dans les classes $\mathcal{C}$ est telle que $|\mathcal{T}_{c_1}| = 10$, $|\mathcal{T}_{c_2}| = 85$, $|\mathcal{T}_{c_3}| = 5$ et $|r| = 100$. Nous considérons deux itemsets dans r : l'itemset X tel que $freq(X, r) = 9$, $freq(X, r_{c_1}) = 7$, $freq(X, r_{c_2}) = 0$, $freq(X, r_{c_3}) = 2$ et l'itemset Y tel que $freq(Y, r) = 45$, $freq(Y, r_{c_1}) = 0$, $freq(Y, r_{c_2}) = 40$, $freq(Y, r_{c_3}) = 5$.*

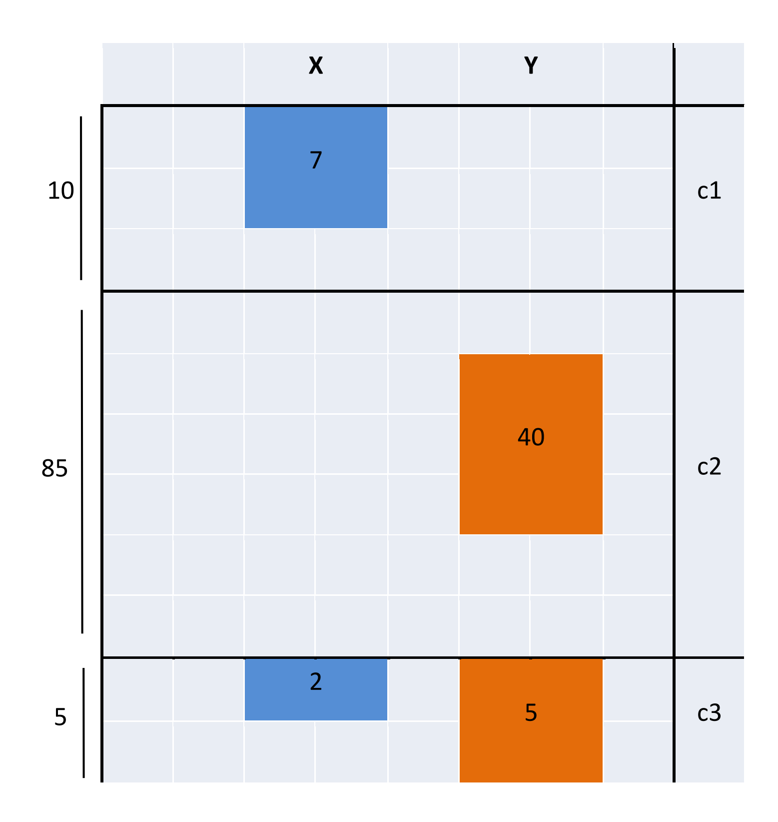


FIGURE 4.1 – Exemple de données aux classes disproportionnées.

Définition 17 (Facteur d'intérêt et corrélation positive) *Soit $\pi : I \rightarrow c$, une règle d'association de classe concluant sur un attribut classe c. Selon [TSK05], le facteur d'intérêt du couple (I, c) dans $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ est défini comme suit :*

$$FInt(I, c, r) = \frac{|\mathcal{T}| \times f_{11}}{f_{1*} \times f_{*1}}$$

$I \to c$	c	$\bar{c}$	Σ
I	f_{11}	f_{10}	f_{1*}
$\bar{I}$	f_{01}	f_{00}	f_{0*}
Σ	$f_{*1} = \mathcal{T}_c$	$f_{*0} = \mathcal{T}_{\bar{c}}$	$f_{**} = \lvert\mathcal{T}\rvert$

TABLE 4.1 – Table de contingence pour la règle d'association $I \to c$ qui conclut sur l'attribut classe c.

Puis,

$$FInt(I,c,r) \begin{cases} = 1, & \textit{si } I \textit{ et } c \textit{ sont indépendants ;} \\ > 1, & \textit{si } I \textit{ et } c \textit{ positivement corrélés ;} \\ < 1, & \textit{si } I \textit{ et } c \textit{ négativement corrélés.} \end{cases}$$

Le cadre fréquence-confiance : Dans l'exemple 1, nous avons $conf(Y \to c_2, r) = 40/50$. Ainsi, $Y \to c_2$ peut être considérée comme une règle à forte confiance pour caractériser la classe c_2. Pourtant, nous avons $TA(Y, r_{c_3}) = (5/5)/(40/95) > 2$ ce qui indique que Y apparaît (relativement) deux fois plus dans la classe c_3 que dans le reste de la base ; Y est donc un motif émergent pour la classe c_3. De plus, nous avons aussi $FInt(Y, c_3, r) = (100 \times 5)/(45 \times 5) > 1$, donc Y est positivement corrélé avec c_3. Par contre, la confiance nous induit en erreur, car le calcul de la confiance de $Y \to c_2$ ne tient pas compte de la taille des classes ni de la répartition des erreurs des règles dans les différentes classes du reste de la base. Notons aussi que, d'une manière générale, si l'on utilise un seuil global de fréquence minimale, alors pour espérer caractériser la classe minoritaire ce seuil devra s'abaisser à l'échelle de la classe minoritaire ; ce qui aura pour effet de créer un biais vers la classe majoritaire – i.e. on obtiendra beaucoup plus de règles pour la classe majoritaire.

Le cadre des itemsets émergents : Si l'on considère l'approche à base de motifs émergents, nous avons $TA(X, r_{c_1}) = (7/10)/(2/90) > 31$. X apparaît donc relativement 31 fois plus dans c_1 que dans le reste de la base et peut être considéré comme un motif qui caractérise c_1. Pourtant, $TA(X, r_{c_3}) = (2/5)/(7/95) > 5$ indiquerait que X est caractéristique de la classe c_3. Le taux d'accroissement, bien que tenant compte d'une certaine manière de la taille de certaines classes, nous induit aussi en erreur et génère des conflits de règles car la répartition des erreurs dans les différentes classes n'est pas prise en compte.

Le cadre des itemsets positivement corrélés : Dans le cadre des itemsets positivement corrélés nous sommes confrontés au même problème que pour les itemsets émergents. En effet, nous avons $FInt(X, c_1, r) = (100 \times 7)/(9 \times 10) > 1$ et $FInt(X, c_3, r) =$

$(100 \times 2)/(9 \times 5) > 1$ qui indiquent que X est positivement corrélés avec les classes c_1 et c_3. Ces conflits sont aussi dus à la répartition des erreurs dans les différentes classes.

Les différents cadres rappelés dans l'exemple précédent suivent une approche dite `OVA` (`One-Versus-All`) car les motifs extraits caractérisent une classe c_i par rapport à l'union des autres classes qui constituent le reste des données. Nous avons montré que les problèmes liés aux différentes mesures d'intérêt (confiance, taux d'accroissement, facteur d'intérêt) et aux conflits de règles sont intrinsèquement liés à l'approche `OVA` , i.e. viennent du fait que soit la taille des classes, soit la fréquence relative des motifs dans chaque classe n'est pas prise en compte. Dans la suite nous proposons un nouveau cadre de travail, appelé `OVE` (`One-Versus-Each`) dans lequel la taille des classes ainsi que la fréquence relative des itemsets dans chaque classe sera considérée.

Une autre manière de traiter les problèmes multi-classes est de suivre une approche dite `OVO` [Für02, WLW04, PF07]. Le principe est de diviser un problème de classification supervisée à n classes en $n(n-1)/2$ sous-problèmes à 2 classes. Chaque sous-problème donne lieu à la construction d'un classifieur. Pour un nouvel exemple t, les différents classifieurs sont combinés pour décider la classe de t. Par exemple, dans [Für02], un simple vote des différents classifieurs est utilisé pour déterminer la classe à prédire pour t. Toutefois, les problèmes aux classes disproportionnées sont identifiés aussi comme une limite aux approches `OVO` (voir [Für02]). En effet, les classifieurs issus des sous-problèmes contenant la classe majoritaire peuvent être biaisés (vers la classe majoritaire). Ce biais sera reporté à la phase de combinaison des classifieurs et la classe majoritaire aura tendance à être la plus prédite.

4.2 Vers une approche `OVE`

Considérons $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ une base de données binaires à p classes $(c_1, c_2, \ldots, c_p \in \mathcal{C})$. Pour tenir compte des différentes tailles de classes $(|\mathcal{T}_{c_1}|, \ldots, |\mathcal{T}_{c_p}|)$ et éviter un biais vers la classe majoritaire, nous devons disposer pour chacune des classes d'un seuil de fréquence minimum à partir duquel un motif extrait sera considéré comme intéressant. Pour prendre en compte la répartition des erreurs (faux positifs) d'un motif caractérisant c_i dans chaque autre classe $c_j \neq c_i$, nous devons aussi disposer d'un seuil de fréquence maximum (définissant une limite d'infréquence) pour chaque c_j. Nous utilisons alors une matrice Γ pour représenter tous ces seuils locaux.

4.2.1 Matrice de seuils et règles de caractérisation OVE

Soit Γ une matrice de seuils de fréquence pour un problème de classification à p classes.

$$\Gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,p} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p,1} & \gamma_{p,2} & \cdots & \gamma_{p,p} \end{pmatrix}$$

Les paramètres d'une ligne i de Γ, sont les seuils de fréquence et d'infréquence pour les motifs X caractérisant la classe c_i. Plus précisément, $\gamma_{i,i}$ (i^{eme} ligne et i^{eme} colonne de Γ) est le seuil de fréquence minimum pour la classe c_i et les $\gamma_{i,j}$ sont les seuils de fréquence maximum pour X dans les classes $c_j (j \neq i)$. Ainsi X sera $\gamma_{i,i}$-fréquent dans r_{c_i} et infréquent dans chacune des autres classes. Nous pouvons maintenant définir les règles de caractérisation OVE comme suit :

Définition 18 (Règle de caractérisation OVE) *Soit Γ une matrice de seuils de fréquence et d'infréquence. La règle d'association $\pi : X \rightarrow c_i$ est une règle de caractérisation OVE par rapport à Γ pour la classe c_i, notée OVE-CR, si et seulement si les trois conditions suivantes sont vérifiées :*

1. *X est fréquent dans r_{c_i}, i.e. $freq_r(X, r_{c_i}) \geq \gamma_{i,i}$*
2. *X est infréquent dans toutes les autres classes, i.e. $\forall j \neq i, freq_r(X, r_{c_j}) < \gamma_{i,j}$*
3. *X le corps de π est minimal, i.e. $\forall Y \subset X, \exists j \neq i \mid freq_r(Y, r_{c_j}) \geq \gamma_{i,j}$*

Si les conditions 1 et 2 nous assurent que les motifs X extraits respectent bien les contraintes de fréquence et d'infréquence imposées par Γ, la condition 3 nous assure la minimalité du corps X de π. Toutefois, pour s'attaquer à un problème de classification supervisée, d'autres contraintes sont nécessaires sur la combinaison des paramètres de Γ.

4.2.2 Contraintes entre paramètres

Contraintes sur les paramètres de Γ

Intuitivement, si l'on veut qu'une OVE-CR $\pi : X \rightarrow c_i$ soit caractéristique de la classe c_i, il faut que sur la i^{eme} ligne de Γ, le seuil de fréquence relative $\gamma_{i,i}$ soit supérieur à chaque autre $\gamma_{i,j}$ de la même ligne – i.e. ainsi les X extraits apparaissent relativement plus souvent dans r_{c_i} que dans chacune des autres classes $c_j (j \neq i)$. D'autre part, soit $\pi' : Y \rightarrow c_j$ une OVE-CR qui caractérise c_j, alors il faut que $\gamma_{j,i}$ (le nombre d'erreurs maximum autorisées pour π' dans c_i) soit inférieur à $\gamma_{i,i}$ – sans quoi Y serait assez fréquent (relativement)

dans r_{c_i} pour être aussi caractéristique de c_i. Nous formalisons ces deux intuitions avec les deux contraintes suivantes :

$$\text{Contrainte ligne : } \mathbb{C}_{ligne} \equiv \forall i \in \{1, \ldots, n\}, \forall j \neq i, \gamma_{i,j} < \gamma_{i,i}$$

$$\text{Contrainte colonne : } \mathbb{C}_{colonne} \equiv \forall i \in \{1, \ldots, n\}, \forall j \neq i, \gamma_{j,i} < \gamma_{i,i}$$

Dans la suite nous montrons que les deux contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$ nous permettent de faire le lien entre OVE-CR, motifs émergents et corrélation. Plus important, elles permettent d'éviter des conflits de corps de règles et de nous assurer la pertinence de nos motifs pour la classification supervisée.

Liens avec les EPs et la corrélation positive

Nous pouvons voir l'extraction des corps de OVE-CRs qui respectent la contrainte $\mathbb{C}_{ligne}$ (par rapport à une matrice Γ de seuils) comme une généralisation de la définition de l'émergence d'un motif en considérant son intérêt par rapport à chaque classe prise individuellement. En effet :

- si $\forall i \in \{1, \ldots, p\}$ et $\forall j \neq i$, $\gamma_{i,i} = \gamma_{j,j}$ et $\gamma_{i,j} = 0$, alors tout corps X d'une OVE-CR $\pi : X \rightarrow c_i$ est un JEP $\gamma_{i,i}$-fréquent.
- si $\exists i, j$ tel que $\gamma_{i,i} \neq \gamma_{j,j}$, alors nous extrayons toujours des JEPs fréquents mais avec des seuils de fréquence différents pour certaines classes.
- si $\forall i \in \{1, \ldots, p\}$ et $\forall j \neq i$, $\gamma_{i,i} = \gamma_{j,j}$ et $\gamma_{i,j} > 0$, alors tout corps X d'une OVE-CR $\pi : X \rightarrow c_i$ est un ρ-EP $\gamma_{i,i}$-fréquent où $\rho = \gamma_{i,i}/(\max_{j \neq i} \gamma_{i,j})$.
- si $\exists i, j$ tel que $\gamma_{i,i} \neq \gamma_{j,j}$, alors nous extrayons toujours des EPs fréquents mais avec des seuils de fréquence et de TA différents pour certaines classes.

Dans tous les cas, les corps X de OVE-CRs qui concluent sur c_i, extraits selon les seuils de la i^{eme} ligne de Γ, ont une fréquence relative plus élevée dans la classe c_i que dans chacune des autres classes. Ceci nous assure que $TA(X, r_{c_i}) > 1$ et donc que X est positivement corrélé avec c_i (voir proposition 1 page 69). Toutefois nous avons vu qu'il est possible qu'un itemset X tel que $TA(X, r_{c_i}) > 1$ soit positivement corrélé avec plusieurs classes – ce qui génère des conflits de corps de règles. Par la proposition 3 (voir la démonstration en appendice), nous montrons que la contrainte $\mathbb{C}_{colonne}$ nous permet d'éviter ce problème de conflits de corps de règles.

Proposition 3 *Soit S l'ensemble des OVE-CRs dont les corps respectent les seuils de fréquence et d'infréquence d'une matrice Γ et les contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$. Alors S est sans conflit de corps de règles, i.e. il n'existe pas deux règles $\pi : X \rightarrow c_i$ et $\pi' : Y \rightarrow c_j$ telles que $Y \subseteq X$ et $j \neq i$.*

Dans la suite, nous décrivons techniquement comment à partir d'une base de données r, extraire l'ensemble $S = \cup_i S_{c_i}$ des OVE-CRs (où S_{c_i} est l'ensemble des OVE-CRs qui

concluent sur la classe c_i) respectant les contraintes de fréquence et d'infréquence d'une matrice Γ donnée et les contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$ et tel que S est sans conflit de corps de règles. Cet ensemble S constituera notre ensemble de règles "intéressantes" pour la classification supervisée.

4.2.3 Extraction

L'extraction des règles de caractérisation `OVE` est effectuée classe par classe. Soit $c_i \in \mathcal{C}$ une classe, l'extraction complète de l'ensemble S_{c_i} des `OVE`-CRs qui concluent sur c_i se fait en fonction des paramètres de la i^{eme} ligne de la matrice de paramètres Γ. $\gamma_{i,i}$ est le seuil de fréquence minimum relatif à la classe c_i et les $\gamma_{i,j}$ sont les seuils de fréquence maximum relatifs aux classes c_j ($j \neq i$). L'espace de recherche des `OVE`-CRs est partiellement ordonné selon $\subseteq$ et a une structure de treillis. Pour extraire S_{c_i}, nous utiliserons un algorithme par niveaux et parcourrons l'espace de recherche en largeur de type `APRIORI` comme décrit dans l'algorithme 9.

Étant donné un niveau k du processus de construction, pour construire un candidat (*child*) de niveau $k+1$, la fonction CONSTRUCTCHILD (ligne 9) teste si le candidat est fréquent dans c_i. Puis, (ligne 12), si le candidat *child* est intéressant (i.e. il est le corps d'une `OVE`-CR) alors il est retenu et ses surensembles sont élagués de l'espace de recherche (`forbiddenPrefixes`) ; sinon (ligne 15) il fera partie des futurs parents pour le niveau d'extraction suivant. De même (ligne 17), si *child* n'est pas fréquent, ses surensembles sont élagués.

L'algorithme 9 nous permet d'obtenir S_{c_i}, l'ensemble des `OVE`CRs pour la classe c_i selon les seuils de fréquence et d'infréquence de la i^{eme} ligne de Γ. Pour obtenir $S = \cup_{c_i \in \mathcal{C}}(S_{c_i})$ l'ensemble de toutes les `OVE`-CRs, on répète l'algorithme d'extraction pour chaque classe de $\mathcal{C}$. Dans la suite, nous proposons une méthode pour agréger les règles de S et prédire la classe d'un nouvel objet o entrant.

D'un point de vue technique, pour calculer les fréquences d'un motif X dans chaque classe c_j (X étant le corps d'une `OVE`-CR concluant sur c_i), nous maintenons p vecteurs de bits (un par classe) pour chaque X. La taille de chacun de ces vecteurs est la même que la taille de la classe à laquelle il est lié. Ainsi, chaque bit indique si l'objet correspondant est couvert par X (=1) ou non (=0) et un simple ET-bit-à-bit rend efficace le calcul des vecteurs de bits des niveaux supérieurs (vecteurs de bits de `child`). De plus, pour stocker les `forbiddenPrefixes`, nous utilisons une structure d'arbre de préfixes qui a déjà fait ses preuves en terme d'efficacité dans les algorithmes par niveaux.

Algorithme 9 : EXTRACT

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$c_i \in \mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ une classe,
$\gamma_{i,i}$ seuil de fréquence minimum pour la classe c_i,
$\gamma_{i,j}$ seuils de fréquences maximum pour les classes $c_j (j \neq i)$

Sortie : S_{c_i} l'ensemble des règles de caractérisation OVE pour c_i

```
forbiddenPrefixes ← ∅;
parents ← [∅];
while parents ≠ [] do
    futureParents ← ∅;
    for all parent ∈ parents do
        forbiddenAtts ← FORBIDDENATTS(forbiddenPrefixes, parent);
        for all attribute > LASTATTRIBUTE(parent) do
            if attribute ∉ forbiddenAtts then
                child ← CONSTRUCTCHILD(parent, attribute);
                if f_{c_i}(child) ≥ γ_{i,i} then
                    if INTERESTING(child) then
                        output (child, c_i);
                        INSERT(child, forbiddenPrefixes);
                    else
                        futureParents ← futureParents ∪ {child};
                else
                    INSERT(child, forbiddenPrefixes);
        parents ← parents \ {parent};
    parents ← futureParents;
```

4.2.4 Classification

Soit $t \in \mathcal{T}$ un nouvel objet entrant, la classe à prédire pour t est déterminée en fonction de l'ensemble $S = \cup_{c_i \in \mathcal{C}}(S_{c_i})$ des OVE-CRs et de leur fréquence relative dans chaque classe. Pour ce faire, pour chaque classe c_i, nous calculons un score $l(t, c_i)$ qui reflète la ressemblance de t avec la classe c_i.

$$l(t, c_i) = \sum_{\{c \in \mathcal{C}\}} \left(\sum_{\{\pi : X \to c \in S | X \subseteq Items(t,r)\}} freq_r(X, r_{c_i}) \right)$$

En fait, pour une classe c_i, $l(t, c_i)$ somme les fréquences relatives (par rapport à r_{c_i}) de toutes les règles que t supporte. Ainsi, les erreurs relatives dans r_{c_i} des règles concluant sur les classes c_j $(j \neq i)$ et supportées par t, sont aussi prises en compte dans le calcul de

$l(t, c_i)$. La classe c_i qui maximise $l(t, c_i)$ est la classe à prédire pour t. Notre implémentation de ce classifieur [CGSB08] est appelé `fitcarc`[1] et son manuel d'utilisation est reporté en appendice.

4.3 Paramétrage automatique avec `fitcare`

Le problème inhérent à notre approche vient du fait que nous devons positionner une multitude de paramètres (en fait exactement p^2 paramètres pour un problème à p classes) contre un ou deux paramètres pour les approches existantes (e.g., fréquence et confiance, fréquence et taux d'accroissement, ...). Si positionner un ou deux paramètres n'est déjà pas chose aisée pour un utilisateur final (éventuellement non-expert en fouille de données), il n'est pas question ici de positionner les paramètres de Γ manuellement. C'est pourquoi, dans cette section, nous proposons une méthode automatique appelée `fitcare`[2] pour paramétrer Γ et qui utilise une approche de recherche locale par hill-climbing (escalade de colline). Ainsi, `fitcare` indiquera automatiquement les "bons" paramètres de Γ pour la tâche de classification supervisée.

4.3.1 Hill-climbing : principe

La méthode Hill-climbing est une technique d'optimisation pour la recherche locale de solution dans un espace de recherche à états discrets. Le but est d'optimiser (i.e. maximiser ou minimiser) une fonction $f(x)$ où les x sont des états discrets. L'espace de recherche est souvent représenté sous forme de graphe où les sommets sont des états de l'espace de recherche et les arcs entre sommets représente la possibilité d'aller d'un état à un autre. La méthode Hill-climbing parcourt ainsi le graphe de sommet en sommet tout en optimisant (i.e. augmentant ou diminuant) $f(x)$, jusqu'à atteindre un optimum local x_m qui est la solution en sortie.

4.3.2 Hill-climbing et `fitcare`

Les points-clés de `fitcare`

La méthode de Hill-climbing est bien adaptée à notre problème de recherche de paramètres et consiste en quatre points-clés que nous détaillerons après :

1. `fitcarc` est l'acronyme récursif en anglais pour : `fitcarc` **i**s **t**he **c**lass **a**ssociation **r**ule **c**lassifier.
2. `fitcare` est l'acronyme récursif en anglais pour : `fitcare` **i**s **t**he **c**lass **a**ssociation **r**ule **e**xtractor.

1. *L'initialisation :* Partant de Γ initialisé à ses paramètres maximaux (respectant $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$), l'étape d'initialisation a pour objectif de trouver de nouveaux paramètres pour Γ par décrémentation. Ces nouveaux paramètres doivent fournir la couverture positive maximale pour chaque classe c_i par l'ensemble S_{c_i} extrait tout en respectant $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$.
2. *La couverture maximale :* La couverture de la base d'apprentissage étant un point clé en classification associative, nous posons la contrainte supplémentaire suivante : "La couverture positive maximale atteinte lors de l'initialisation doit être maintenue" et ce tout au long de la phase d'optimisation.
3. *La fonction à optimiser :* le choix de la fonction à optimiser tout au long de l'optimisation est cruciale. Nous proposerons une mesure d'évaluation pour un ensemble de règles S extrait par rapport à Γ. Plus cette mesure sera grande plus S sera considéré comme meilleur. Nous devrons donc maximiser cette mesure.
4. *Le choix du paramètre à diminuer :* le paramètre à diminuer que nous choisirons sera le plus prometteur pour améliorer la mesure à optimiser.

Initialisation

L'initialisation sera réalisée en trois étapes successives.

Étape 1 - Initialisation de départ : Puisque nous sommes amener à diminuer les paramètres de Γ, alors au départ, tous les paramètres de Γ sont à leur maximum.

$$\forall j \in 1 \dots n, \gamma_{i,j} = \begin{cases} |\mathcal{T}_{c_j}| & \text{si} \quad i = j \\ |\mathcal{T}_{c_j}| - 1 & \text{sinon} \end{cases}$$

Nos premières définitions de Γ et des OVE-CRs (voir définition 18) utilise des seuils relatifs. L'initialisation ainsi que les autres étapes sont présentées avec des seuils absolus car la diminution des de certains paramètres se fait de 1 en 1 (i.e. correspondant à 1 objet). La conversion de fréquence absolue en fréquence relative est triviale et ne change rien au bon déroulement des différentes étapes.

Étape 2 - Recherche de la couverture maximale : pour chaque classe c_i, en partant de la i^{eme} ligne de paramètres, `fitcare` recherche les paramètres qui procurent le meilleur taux de couverture possible pour T_{c_i}. Cette recherche se fait en diminuant $\gamma_{i,i}$ de la plus petite unité possible (i.e. 1 objet en absolu) tant que la couverture maximale n'est pas atteinte et s'arrête dès qu'elle est atteinte. Noter que pour respecter la contrainte $\mathbb{C}_{ligne}$, les $\gamma_{i,j}$ sont diminués en conséquence. A chaque fois que $\gamma_{i,i}$ est diminué il y a extraction avec l'algorithme 9. Dans l'idéal, le meilleur taux à atteindre est 100%. Toutefois, dans certaines

bases de données contenant du bruit de classe (i.e. certains objets sont mal classés), il peut se révéler impossible de couvrir entièrement une classe c_i tout en respectant $\mathbb{C}_{ligne}$. C'est pourquoi, dans ces cas-là, lors de cette étape, on pourra décider d'un taux de couverture inférieur à 100% – en pratique on gardera le meilleur taux de couverture rencontré lors de la descente de $\gamma_{i,i}$. Ainsi, les seuils de fréquence utilisés pour atteindre la plus grande couverture de c_i sont les seuils de Γ à la sortie de cette étape.

Étape 3 - Respect des contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$: Après la recherche de couverture maximale pour chaque classe c_i, chaque ligne de Γ respecte la contrainte $\mathbb{C}_{ligne}$. Toutefois il n'est pas garanti que $\mathbb{C}_{colonne}$ soit respectée puisque chaque ligne de Γ a été traitée indépendamment. Dans cette étape, pour respecter $\mathbb{C}_{colonne}$, nous procédons à des modifications de Γ ligne par ligne. Pour une ligne i de Γ, toute mauvaise valeur de $\gamma_{i,j}$ (i.e. supérieur à $\gamma_{j,j}$) est diminué a minima pour satisfaire $\mathbb{C}_{colonne}$. Puis, une extraction est effectuée avec cette nouvelle ligne de paramètres. Si le taux de couverture de T_{c_i} n'est plus le même que celui obtenu à l'étape 2, alors $\gamma_{i,i}$ est diminué de l'équivalent de 1 objet et une extraction est effectuée jusqu'à ce que le taux de couverture maximal de c_i soit retrouvé (si besoin est, pour respecter $\mathbb{C}_{ligne}$, les $\gamma_{i,j}$ sont diminués en conséquence). Si l'ancien taux de couverture maximal ne peut être atteint, les paramètres apportant le meilleur taux sont retenus. Il en est ainsi pour chaque ligne, et tant qu'une ligne contient des mauvaises valeurs de $\gamma_{i,j}$, c'est-à-dire jusqu'à ce que $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$ soient respectées. A la fin de cette étape, Γ constitue une première solution à notre problème de recherche automatique de paramètres. La dernière extraction avec Γ nous fournit un ensemble S de `OVE`-CRs respectant toutes les contraintes énoncées précédemment et qui constitue notre classifieur.

Hill-climbing contraint par le taux de couverture

Comme évoqué précédemment, bien que dirigé par une fonction à optimiser, l'étape d'optimisation de `fitcare` sera premièrement contraint par le taux de couverture positive maximal atteint pour chaque classe lors de l'initialisation. Ainsi, lors de la phase d'optimisation, si une diminution du paramètre le plus prometteur nous amène à une impossibilité de maintenir la couverture positive maximale, alors la ligne modifiée réaffectée à son ancienne valeur et la diminution du deuxième paramètre le plus prometteur est étudiée.

Maximiser le taux d'accroissement global

Intuitivement, la fonction à optimiser doit nous promettre l'évolution vers un état meilleur que l'état courant dans le processus d'hill-climbing. Un état sera une instance de

Γ et nous modifierons des seuils de Γ pour passer d'un état à un autre. A une instance de Γ correspond un ensemble S d'`OVE`-CRs extraits en fonction de Γ. Comme notre classifieur (ainsi que ses performances) est basé sur S, la fonction à optimiser que nous choisirons sera basée sur S. Une première vision naïve serait de considérer la maximisation du nombre d'objets d'entraînement bien classés par S. Toutefois, *(i)* toute l'information sur l'intérêt des règles est perdue, *(ii)* le fait qu'un objet soit couvert par plusieurs règles n'est pas pris en compte et *(iii)* maximiser la précision sur la base d'entraînement peut conduire à un sur-apprentissage.
Dans notre approche `fitcare` , nous proposons de maximiser les taux d'accroissement globaux. Étant données deux classes $(c_i, c_j) \in \mathcal{C}^2$, le taux d'accroissement global $g(c_i, cj)$ mesure la confusion avec c_j lorsqu'on classifie les objets de $\mathcal{T}_{c_i}$ et est défini comme suit :

$$g(c_i, c_j) = \frac{\sum_{t \in \mathcal{T}_{c_i}} l(t, c_i)}{\sum_{t \in \mathcal{T}_{c_i}} l(t, c_j)}$$

Ainsi, plus $g(c_i, c_j)$ est grand, moins il y a de confusion. Cette mesure a l'avantage de prendre en compte l'intérêt de toutes les règles de S supportées par des objets de $\mathcal{T}_{c_i}$ et aussi de mettre en rapport les ressemblances des objets de $\mathcal{T}_{c_i}$ avec les classes c_i et c_j.

A partir de S extrait en fonction de Γ, `fitcare` calcule toutes les confusions entre les classes, soit n^2-n confusions. `fitcare` tente de maximiser le taux minimal d'accroissement global. Si aucune amélioration n'est possible, `fitcare` tente de maximiser le deuxième plus petit taux (sans diminuer le plus petit), et ainsi de suite...

Choisir le bon paramètre à modifier

Soit $g(c_i, c_j)$ la plus grande confusion (i.e. le taux d'accroissement global minimal de c_i par rapport à c_j). $g(c_i, c_j)$ est donc la mesure à optimiser. Pour espérer améliorer $g(c_i, c_j)$, il nous faut diminuer un paramètre $\gamma_{i,k} (i \neq k)$ ou $\gamma_{j,k} (j \neq k)$ qui sont des causes de la faiblesse de $g(c_i, c_j)$. Pour déterminer la cause principale, chaque classe mise en jeu dans

le dénominateur de $g(c_i, c_j)$ est évaluée séparément :

$$\begin{pmatrix} \sum_{t \in \mathcal{T}_{c_i}} \sum_{\{\pi: X \rightarrow c_1 \in S | X \subseteq Items(t,r)\}} freq_r(X, r_{c_j}) \\ + \\ \sum_{t \in \mathcal{T}_{c_i}} \sum_{\{\pi: X \rightarrow c_2 \in S | X \subseteq Items(t,r)\}} freq_r(X, r_{c_j}) \\ + \\ \vdots \\ + \\ \sum_{t \in \mathcal{T}_{c_i}} \sum_{\{\pi: X \rightarrow c_p \in S | X \subseteq Items(t,r)\}} freq_r(X, r_{c_j}) \end{pmatrix}$$

Nous pensons que le plus grand terme du dénominateur en est la cause principale. En général, le plus grand terme est le i^{eme} terme ou le j^{eme} terme. Le i^{eme} indique que les règles concluant sur c_i sont trop fréquentes dans r_{c_j}, le j^{eme} au contraire indique que les règles concluant sur c_j sont trop fréquentes dans r_{c_i}. Ce terme nous dévoile le paramètre de Γ à diminuer et qui est le plus prometteur. En effet, si le i^{eme} terme est le plus grand, alors $\gamma_{i,j}$ sera diminué (de 1 en absolu) ; si c'est le j^{eme} qui est le plus grand alors $\gamma_{j,i}$ sera diminué.

Une fois qu'un paramètre $\gamma_{i,j}$ (ou $\gamma_{j,i}$) a été diminué, si la contrainte de couverture positive maximale n'est plus respectée, alors il faut diminuer les $\gamma_{i,i}$ (ou $\gamma_{j,j}$) jusqu'à retrouver la couverture maximale et ensuite diminuer les $\gamma_{i,j}$ ou $\gamma_{j,i}$ tout en gardant la couverture maximale. Une nouvelle extraction est alors effectuée et si $g(c_i, c_j)$ est amélioré alors Γ est gardé comme nouvelle meilleure matrice de paramètres ; sinon Γ garde sa meilleure précédente valeur respectant toutes les contraintes et le deuxième paramètre le plus prometteur (issu de la deuxième plus grande confusion) est diminué ; et ainsi de suite…

`fitcare` : un algorithme d'optimisation

Le pseudo-code de notre approche `fitcare` reprenant et mettant en forme les différents points-clés énoncés est reporté dans l'algorithme 10. A la ligne 1, Γ_{best} est initialisé avec la meilleure matrice de paramètres issue de la phase d'initialisation en trois étapes présentée précédemment. Puis la variable `isParametersModified` indiquant si un paramètre de Γ a été modifié, est initialisé à faux et la classe en cours de traitement `classId`, qui indique la classe sur laquelle on extrait les règles (la ligne courante de Γ qui est traitée), est initialisée à la dernière classe.

La principale boucle (ligne 4) qui constitue la partie optimisation de notre approche `fitcare` prend fin lorsque la classe à traiter (`classId`) dépasse p.

Premier tour : au premier tour de boucle il ne passe rien car les paramètres ne sont pas modifiés. Au test de validation (ligne 17), Γ_{best} est initialisé à Γ et enfin (ligne 21), la diminution des paramètres (et donc modification de Γ) commence et indique la classe pour le prochain tour de boucle. En fait, l'initialisation de `classId` à p force le passage de la condition (ligne 16) pour lancer une évaluation et ainsi, évite p tours de boucle à vide.

Optimisation : après la première itération, lors d'une itération k de la boucle principale, si la ligne de paramètres pour la classe courante `classId` n'a pas été modifiée, la fonction RATIONALIZEPARAMETERS force la ligne courante à respecter les contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$ en diminuant les paramètres (seuils d'infréquence) et renvoie vrai si modification il y a eu sinon faux (lignes 5-6). Une bonne partie de notre espace de recherche est donc élagué.
Puis, on teste si les paramètres ont été modifiés lors de la rationalisation ou à la ligne 21 lors de l'itération précédente.

- Si les paramètres n'ont pas été modifiés, on passe à la classe suivante (ligne 15) et lorsque `classId` $> p$, on teste si Γ nous fournit un meilleur ensemble de règles. Puis on continue la diminution de paramètres (ligne 21).
- Si les paramètres ont été modifiés, alors souvent, la couverture maximale pour la classe courante a été perdue. La fonction LEARNPARAMETERS (ligne 8) tente de retrouver cette couverture maximale en diminuant les paramètres (i.e. en diminuant les seuils de fréquence et si nécessaires en diminuant les seuils d'infréquence pour respecter $\mathbb{C}_{ligne}$). Si la couverture maximale est retrouvée, alors les seuils d'infréquence sont diminués jusqu'à leur minimum tout en gardant la couverture maximale et LEARNPARAMETERS renvoie vrai – le prochain tour de boucle se fera avec la première classe afin de garantir $\mathbb{C}_{colonne}$. Si la couverture maximale est perdue, alors LEARNPARAMETERS renvoie faux et on revient à la meilleure matrice Γ permettant une couverture maximale avant de déterminer le paramètre le plus prometteur à diminuer et repartir pour un tour de boucle avec un nouveau `classId` (lignes 12-13).

Validation : pour valider une nouvelle matrice Γ de paramètres (ligne 17), on teste si elle est meilleure que la matrice Γ_{best} courante. Au lieu de comparer la précision des classifieurs issus des règles extraites en fonction de Γ et Γ_{best} sur toute la base d'apprentissage, on réserve une partie de la base d'apprentissage à cet effet (voir l'option `-t` dans le manuel de `fitcare` en appendice). La matrice de paramètres permettant la meilleure précision sur cette partie test est gardée comme nouvelle meilleure matrice.

Algorithme 10 : `fitcare`

Entrée : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$ un contexte binaire,
$\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ l'ensemble des classes

Sortie : Γ_{best} la meilleure matrice de paramètres-seuils de fréquence et d'infréquence

```
Γ ← INIT(r);
isParametersModified ← false;
classId ← p;
while classId ≤ p do
    if ¬ isParametersModified then
        isParametersModified ← RATIONALIZEPARAMETERS(classId);
    if isParametersModified then
        if LEARNPARAMETERS(classId) then
            isParametersModified ← false;
            classId ← 1 ;
        else
            Γ ← Γbest;
            classId ← DECREASEMOSTPROLEMATICDELTA();
    else
        classId ← classId + 1;
        if classId > p then
            if BETTERVALIDATION() then
                Γbest ← Γ;
            else
                Γ ← Γbest;
            classId ← DECREASEMOSTPROBLEMATICDELTA();
            isParametersModified ← true;
return Γbest
```

4.4 Validation expérimentale

Implémentation

L'extracteur de règles de caractérisation `OVE` , l'algorithme `fitcare` [CGSB08] ainsi que le classifieur à base de règles `fitcarc` ont été implémentés en `C++`[3]. Bien que `fitcare` soit libre de tout paramétrage, il est toutefois possible de jouer avec certains paramètres

3. `fitcare` et `fitcarc` ont été implémentés par Loïc Cerf au LIRIS.

(couverture, fréquence, matrice de fréquences,...) à la demande de l'utilisateur. Pour plus de détails sur les options de `fitcare` et de `fitcarc`, leurs manuels (pages man) sont reportés en appendice.

Expérimentations et protocole

Pour valider l'approche `fitcare` , nous l'expérimentons sur des données aux caractéristiques différentes : *(i)* des données à deux classes, *(ii)* des données multi-classes, *(iii)* des données aux classes disproportionnées. Toutes les bases utilisées sont issues du répertoire UCI [AN07] – à l'exception de la base `meningite` fournit par P. François et B. Crémilleux. Lorsque les données contiennent des attributs continus, la base d'apprentissage est discrétisée selon la méthode de Fayyad et Irani [FI93], puis binarisée. Le schéma de binarisation obtenu est reporté sur la base test. Les résultats de précision présentés plus loin sont obtenus par 10-CV. Nous comparons nos résultats de précision avec ceux de CPAR et HARMONY . Pour une comparaison honnête, les résultats de précision de `fitcare` , CPAR et HARMONY sont obtenus avec la même discrétisation/binarisation et avec la même validation croisée (générée par WEKA). Pour ce faire, nous utilisons l'implémentation de CPAR réalisée en JAVA par [Coe04] que nous modifions légèrement pour produire des résultats de précision par classe et l'implémentation de HARMONY réalisée en C++ par ses auteurs [WK05, WK06].

Résultats de précision et comparaison

Tous les résultats de précision sont reportés dans la table 4.2. La première colonne (*"Données"*) indique la répartition des classes pour chaque base utilisée. Lorsqu'une classe c_i est telle que $|\mathcal{T}_{c_i}|/|\mathcal{T}_{c_{max}}| \leq 0,6$ où c_{max} est la classe majoritaire, alors nous considérons que nous sommes face à un problème aux classes disproportionnées et c_i est une des classes minoritaires (mise en gras). Dans les autres colonnes, nous reportons la précision globale et la précision par classe pour chaque classifieur.

Au vu des résultats de précision globale, HARMONY semble le plus performant. En effet, HARMONY obtient 11 fois sur 19 le meilleur score (ou un des meilleurs), alors que `fitcare` obtient 8 et CPAR seulement 3. Pour les comparaisons deux à deux *(gagné-nul-perdu)*, les résultats sont les suivants : `fitcare` vs CPAR 14-1-4 et `fitcare` vs HARMONY 6-2-11. En terme de précision globale, HARMONY est meilleur que `fitcare` .

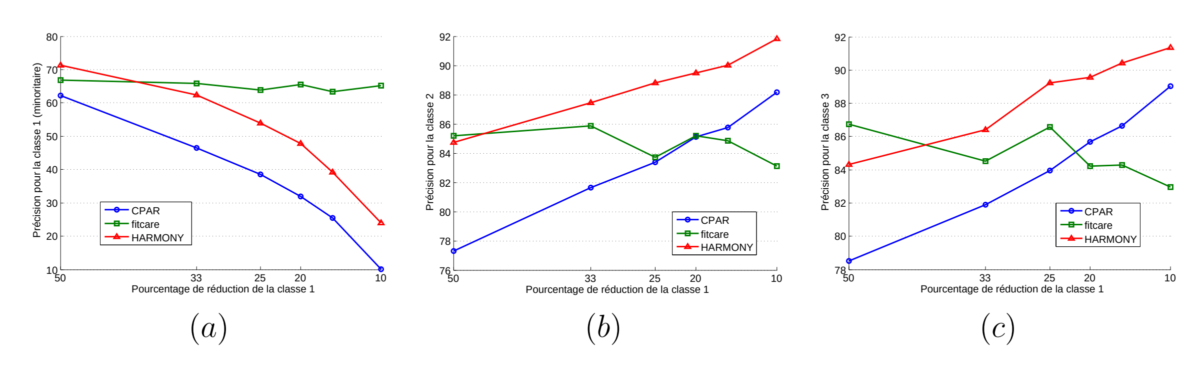
Toutefois lorsque les classes sont disproportionnées, ce sont souvent les résultats de précision sur la ou les classes minoritaires qui sont importants. Les résultats de précision par classe (voir résultats en gras en table 4.2) donne `fitcare` gagnant. En effet, `fitcare` obtient 14 fois sur 19 la meilleure précision (ou une des meilleures) sur les classes minoritaires, alors que CPAR et HARMONY obtiennent seulement 5 meilleurs résultats. Les

résultats de comparaisons deux à deux sont 13-3-3 pour `fitcare` vs `CPAR` et 12-4-3 pour `fitcare` vs `HARMONY` . Clairement, `fitcare` est plus performant en terme de précision sur les classes minoritaires que `CPAR` et `HARMONY` . Notons aussi que pour les bases `diabetes`, `hepatitis`, `labor` et `meningite`, alors que `fitcare` réalise un meilleur score de précision sur la classe minoritaire, `CPAR` et `HARMONY` quant à eux obtiennent de bien meilleurs scores de précision sur la classe majoritaire – ce qui tend à confirmer l'hypothèse concernant le biais vers la classe majoritaire, énoncée en début de chapitre. Dans la suite, nous menons d'autres expériences sur des bases artificiellement modifiées pour étudier les effets de la disproportion des classes sur les résultats des différents classifieurs.

Evolution de la précision par rapport à la disproportion des classes : pour avoir une meilleure idée du comportement de `CPAR` , `fitcare` et `HARMONY` face à des problèmes multi-classes disproportionnées, nous expérimentons les trois classifieurs sur la base de données `waveform` dont la répartition des classes est modifiée artificiellement. `waveform` est une base de données à trois classes telle que $|\mathcal{T}_{c_1}| = 1657$, $|\mathcal{T}_{c_2}| = 1647$ et $|\mathcal{T}_{c_3}| = 1696$ (originalement bien proportionnée). A partir de la base originale, nous construisons diverses bases dans lesquelles une ou deux classes (c_i, c_j) sont artificiellement disproportionnées. Pour ce faire, nous partitionnons les objets de la classe concernée (c_i) en x sous-ensembles d'à peu près la même taille : $x \in \{2, 3, 4, 5, 6, 10\}$. La classe c_i se trouve alors réduite à $y = 50\%$, 33%, 25%, 20%, 16% ou 10% de sa taille originale. Une nouvelle base ainsi construite est composée d'un sous-ensemble des objets de classe c_i et de tous les objets de classe c_j $(j \neq i)$.
Les graphiques des figures 4.2, 4.3 et 4.4 rapportent l'évolution de la précision par classe de `CPAR` , `fitcare` et `HARMONY` sur les versions de `waveform` avec des classes disproportionnées. Les résultats de précision pour une version de `waveform` dont la classe c_i a été réduite à $y\%$ sont obtenus en moyennant les x résultats de précision (obtenus par 10-CV) sur les sous-ensembles correspondants. Par exemple, lorsque la classe c_1 est réduite à 50% (respectivement 33%, 25%,...), la précision reportée dans les graphiques de la figure 4.2 est la moyenne des 2 (respectivement 3, 4,...) précisions obtenues (par 10-CV) sur les 2 (respectivement 3, 4,...) sous-ensembles de données.

Dans les graphiques de la figure 4.2, seule c_1 est réduite et est donc la classe minoritaire. On voit très bien que plus elle est réduite, plus la précision de `CPAR` et `HARMONY` sur c_1 diminue (voir graphique a). Cette diminution est notable puisque la précision sur c_1 chute jusqu'à 10% pour `CPAR` et 25% pour `HARMONY` . Au contraire, la précision de `fitcare` sur c_1 est beaucoup plus stable lors de la réduction de la taille de c_1. Dans les graphiques b et c, nous nous intéressons à la précision des classifieurs sur les classes majoritaires c_2 et c_3 lorsque c_1 est réduite. Plus c_1 est réduite, plus la précision de `CPAR` et `HARMONY` augmente pour les classes majoritaires. Au contraire, la précision de `fitcare` sur les classes majoritaires diminue légèrement avec la réduction de la taille de c_1.

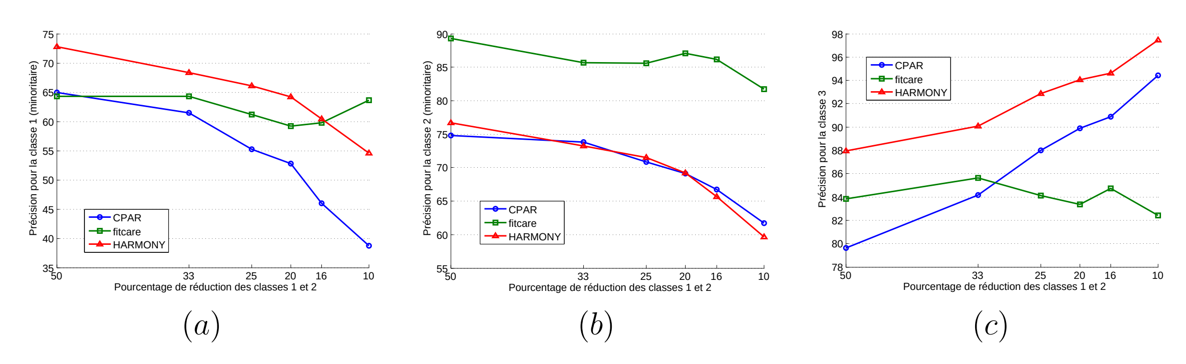
Dans les graphiques des figures 4.3, 4.4 et 4.5, deux classes sont réduites selon le même



(a) (b) (c)

FIGURE 4.2 – Evolution de la précision par classe lorsque la classe 1 est minoritaire pour `CPAR` , `fitcare` et `HARMONY` sur la base `waveform`.

ratio et sont donc minoritaires et nous avons donc une classe majoritaire. Nous retrouvons les mêmes observations que précédemment : la réduction de la taille des classes implique une chute de précision pour `CPAR` et `HARMONY` sur les classes concernées (minoritaires) et une augmentation de la précision sur la classe majoritaire. Au contraire, la prédiction de `fitcare` sur les classes minoritaires reste stable face à la réduction des classes et diminue légèrement pour la classe majoritaire.



(a) (b) (c)

FIGURE 4.3 – Evolution de la précision par classe lorsque les classes 1 et 2 sont minoritaires pour `CPAR` , `fitcare` et `HARMONY` sur la base `waveform`.

Notre étude expérimentale de l'évolution de la précision par classe des approches `OVA` comme `CPAR` et `HARMONY` dans les données aux classes disproportionnées confirme les problèmes énoncés en début de chapitre ; à savoir un biais des approches `OVA` vers la classe majoritaire et donc une détérioration de la précision sur la classe minoritaire. Notre approche `fitcare` suivant une approche `OVE` offre une solution pour pallier ce problème. En effet, en tenant compte de la répartition des classes et la répartition des erreurs des règles extraites dans les différentes classes, `fitcare` évite le biais vers la classe majoritaire et obtient des résultats de précision bien supérieurs aux approches `OVA` comme `CPAR` ou `HARMONY` pour la classe minoritaire.

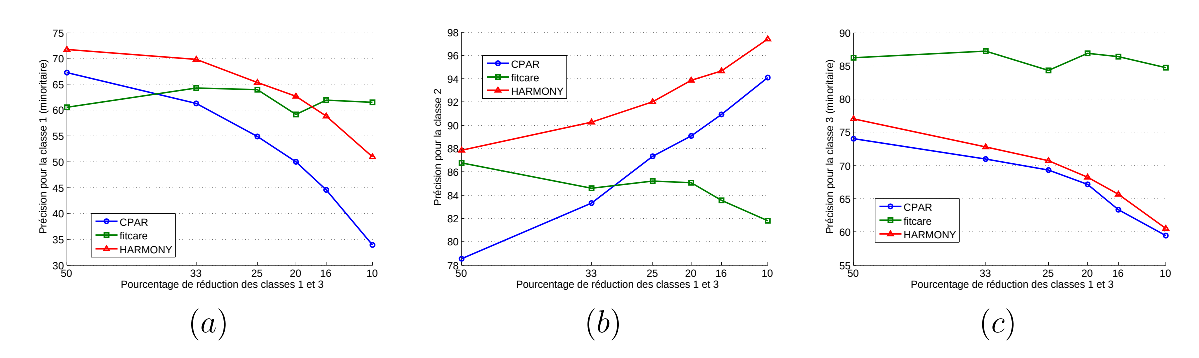


FIGURE 4.4 – Evolution de la précision par classe lorsque les classes 1 et 3 sont minoritaires pour `CPAR` , `fitcare` et `HARMONY` sur la base `waveform`.

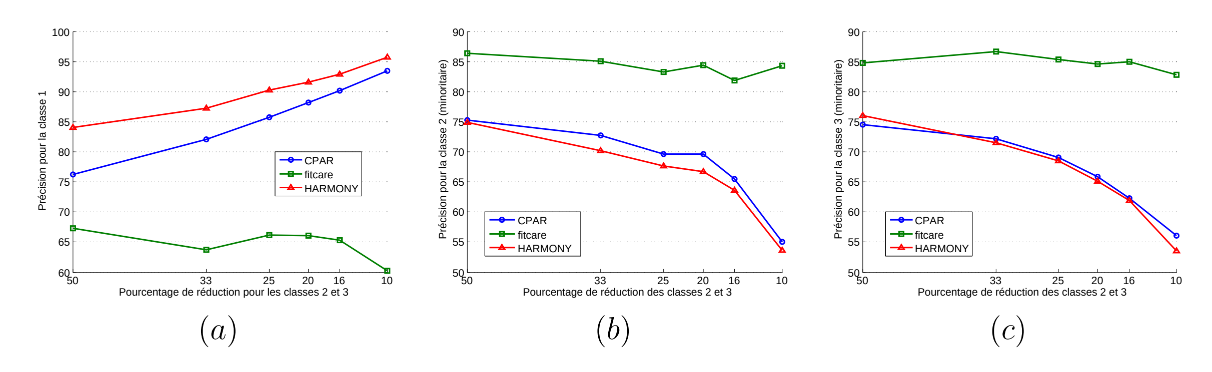


FIGURE 4.5 – Evolution de la précision par classe lorsque les classes 2 et 3 sont minoritaires pour `CPAR` , `fitcare` et `HARMONY` sur la base `waveform`.

4.5 Discussion et limites

Concernant les trois points-clés de la classification associative énoncés en introduction, notre approche `fitcare` propose les solutions suivantes : la définition des `OVE`-CRs indique que les règles extraites ont un corps minimal et que l'ensemble $S_{\gamma,\delta}$ des `OVE`-CRs extraites est exempt de redondance. Le respect des contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$ sur Γ nous assurent que le corps des `OVE`-CRs extraites ont le pouvoir discriminant des motifs émergents tout en évitant des conflits de règles dans $S_{\gamma,\delta}$. Ces contraintes prennent aussi en compte la distribution des classes et la fréquence relative des corps de règles dans chacune des classes. Enfin, tout au long de l'optimisation, `fitcare` maintient le taux de couverture maximal obtenu lors de l'initialisation, ce qui nous assure un taux de couverture satisfaisant.

Les résultats expérimentaux ont montré que l'approche `OVE` `fitcare` est bien plus performante en terme de précision sur les classes minoritaires dans les contextes multi-classes disproportionnées que des approches `OVA` comme `CPAR` et `HARMONY` . De plus, `fitcare` évite le biais vers la classe majoritaire que subissent `CPAR` et `HARMONY` .

Notons toutefois la principale limitation de `fitcare` : dans certains cas, l'approche par optimisation des paramètres de Γ peut être très gourmande en temps de calcul. En effet, pour des bases de données de taille conséquente (avec un grand nombre d'attributs), l'extraction des `OVE`-CRs pour des seuils de fréquence très bas est difficile. Si de plus la base contient un grand nombre d'objets, `fitcare` devra réaliser plusieurs de ces extractions difficiles. Pour cette raison, le temps d'exécution de `fitcare` est bien supérieur à celui des approches comme `CPAR` ou `HARMONY` . Notons aussi que si Γ est optimal selon `fitcare` pour des seuils de fréquence très bas, alors le nombre d'`OVE`-CRs peut être très grand. Pour à la fois réduire le temps de calcul et le nombre de règles extraites, il faudrait trouver un compromis entre l'optimalité de Γ et les $\gamma_{i,i}$ (afin peut-être de stopper le processus d'optimisation avant terme). Cela des demande des investigations plus approfondies.

Données	*Précision globale*			*Précision par classe (taux de vrais positifs)*		
	CPAR	fitcare	HARMONY	CPAR	fitcare	HARMONY
balance-scale 288/**49**/288	70,08	**75,04**	73,12	66,52/3/83,1	79,86/**4**/82,29	80,55/0/78,12
breast-cancer 201/**85**	**70,63**	66,08	69,93	78,15/**61,95**	70,64/55,29	78,1/50,58
breast-w 458/**241**	94,14	**96,70**	95,70	94,48/94,04	96,72/**96,68**	97,37/92,53
car 1210/**384**/**69**/**65**	78,42	83,85	**89,35**	91,52/54,16/39,34/37,05	92,39/61,71/**66,67**/**73,84**	95,37/**85,67**/24,63/67,69
colic 232/**136**	81,25	81,79	**82,88**	84,35/**79,31**	84,91/76,47	89,65/71,32
credit-a 307/383	85,51	81,01	**85,65**	79,03/93,01	79,47/82,24	81,1/89,29
diabetes 500/**268**	**73,31**	64,45	73,04	<u>77,23</u>/64,42	58,6/**75,37**	<u>80,2</u>/59,7
heart-c 165/138	78,82	**80,52**	78,87	80,79/79,6	80/81,15	81,21/76,08
heart-h 188/**106**	78,3	78,57	**82,31**	81,64/**75,96**	85,63/66,03	90,42/67,92
heart-s 150/120	81,48	**83,33**	81,48	83,52/81,92	84,66/81,66	84,66/77,5
hepatitis **32**/123	78,54	79,35	**85,16**	56,59/<u>92,93</u>	**84,37**/78,04	50/<u>94,3</u>
iris 50/50/50	**94,67**	**94,67**	**94,67**	100/92,67/92,67	100/90/94	100/92/92
labor **20**/37	68,67	**80,7**	**80,7**	54,83/<u>85,17</u>	**85**/78,37	75/<u>83,78</u>
meningite **84**/245	87,51	**93,61**	92,7	71,46/<u>95,59</u>	**88,09**/95,51	72,61/<u>99,59</u>
sonar 97/111	75,48	76,44	**81,73**	70,1/87,37	76,28/76,57	83,5/80,18
ticTacToe 626/**332**	70,98	89,87	**97,18**	77,08/60,03	88,17/93,07	99,2/**93,37**
waveform 1657/1647/1696	74,28	79,74	**80,3**	71,28/75,99/75,8	66,84/87,77/84,89	77,71/81,24/81,99
wine 59/71/48	93,86	91,57	**96,06**	92,14/93,67/98,33	96,61/83,09/97,91	96,61/95,77/95,83
zoo 41/**20**/**5**/**13**/**4**/**8**/**10**	93,18	**95,04**	92,07	94,67/**100**/20/**100**/30/55/80	97,56/**100**/**60**/**100**/**75**/**100**/**90**	97,56/**100**/40/92,3/**75**/87,5/**90**

TABLE 4.2 – Résultats de précision de `fitcare` et comparaison avec `CPAR` et `HARMONY` .

Quatrième partie

Scénario de découverte de connaissances appliqué à l'érosion des sols en Nouvelle-Calédonie

Chapitre 5

Caractérisation de l'érosion des sols en Nouvelle-Calédonie

Sommaire

5.1 Contexte général

La Nouvelle-Calédonie est un archipel français situé dans le pacifique sud-ouest à 1500 kilomètres à l'est de l'Australie. C'est aussi une des zones dans le monde à être considérée comme "hot-spot" pour la bio-diversité. En 2008 les 2/3 des $24000 km^2$ du lagon néo-calédonien ont été ajoutés à la liste du Patrimoine mondial de l'UNESCO. La gestion et la protection de l'environnement est un enjeu majeur en Nouvelle-Calédonie. Notons aussi que la Nouvelle-Calédonie possède à elle seule 1/4 des réserves de nickel de la planète. Forte de ses trois industries minières de grande envergure, elle est ainsi le 5ème producteur mondial de nickel. La présence de trois grands projets miniers sur le territoire nécessite une approche globale de la surveillance de l'environnement. Divers travaux sur les thèmes

du changement climatique et de l'impact anthropique sur l'environnement sont financés par des plans régionaux et nationaux.

5.1.1 Problématique de l'érosion

Dans ce contexte, une des thématiques principales étudiées au sein du PPME (Pôle Pluridisciplinaire de la Matière et de l'Environnement) est l'érosion des sols. En Nouvelle-Calédonie, l'érosion des sols a un impact fort et global sur les écosystèmes terrestres et côtiers (des montagnes, en passant par les plaines alluviales et les mangroves, jusqu'au lagon et aux barrières de corail). Feux de brousse, déforestation et activités humaines sont des accélérateurs du processus d'érosion en montagne. Les sédiments ainsi produits sont transportés vers les plaines côtières, la mer et le lagon via les principales lignes de drain. L'impact sur les activités humaines telles que les mines à ciel ouvert, l'élevage ou la pêche est immédiat et récurrent. Il est donc important d'identifier les facteurs des processus d'érosion afin de planifier une gestion durable de l'environnement. Plus précisément, une estimation de l'aléa érosion [1] est nécessaire pour assurer des nouveaux développements avec des impacts réduits de l'érosion. Cependant, la cartographie des zones d'érosion ainsi que l'estimation de l'aléa érosion à l'échelle de région sont coûteuses en temps et en argent et sont rarement mises à jour. Les données disponibles pour tenter de décrire le phénomène d'érosion sont volumineuses et hétérogènes (images satellitaires, mesures physiques, observations qualitatives,. . .). Ainsi, pour comprendre et prédire ce phénomène environnemental, des techniques d'analyse avancées et (semi)-automatique sont nécessaires.

Dans la suite, nous décrivons les données d'érosion mises à notre disposition et utilisées dans nos expérimentations. Puis, nous proposons un scénario complet de découverte de connaissances afin d'obtenir une première caractérisation du phénomène d'érosion ainsi qu'une première estimation de l'aléa érosion sur les zones étudiées. Ainsi, nous pourrons produire une cartographie des zones érodées et de l'aléa érosion. Les phénomènes d'érosion apparaissant sur une petite partie des sols des zones étudiées (environ 3%), nous sommes face à une base de données à 2 classes (sol érodé / sol non-érodé) disproportionnées et `fitcare` sera au coeur de notre scénario de découverte de connaissances.

5.1.2 Bases de données sur l'érosion

La zone d'étude est constitué de trois bassins versants limitrophes : celui, de la Ouenghi, de la Tontouta et de Dumbéa (voir la carte des altitude dans la figure 5.1). Un bassin versant est une zone géographique délimitée par une ligne de crête et dont tous les écoulements se dirigent vers le même exutoire (ici la mer). La zone d'étude a été divisée ainsi car le bassin versant est une entité significative pour les géologues experts du

1. aléa érosion : probabilité d'apparition des phénomènes d'érosion

domaine. Pour chaque bassin versant, les experts ont identifié six paramètres physiques comme étant des facteurs du phénomène d'érosion. Ces paramètres sont représentés en couches thématiques de données :

1. *Pluviométrie :* données météorologiques quantifiant les précipitations par extrapolation (valeurs en mm/an moyennées sur 30 ans).
2. *Lithologie :* données de terrain relatant la nature du sol (e.g. latérites minces, épaisses, serpentinites, dunites,...)
3. *Altitude :* données sur l'altitude issues de mesures physiques et du modèle numérique de terrain (MNT) avec une précision à 30m (valeurs en mètres, de 0 à 1500m).
4. *Occupation du sol :* données sur le type de végétation occupant le sol (e.g. maquis minier, savane herbeuse, mangrove clairsemée, dense,...)
5. *Pente :* données sur la pente (inclinaison) du sol issues du modèle numérique de terrain.
6. *Erosion :* c'est la couche classe (sol érodé / sol non-érodé) issue des valeurs calculées d'indice de brillance sur des photographies satellitaires `SPOT 5`.

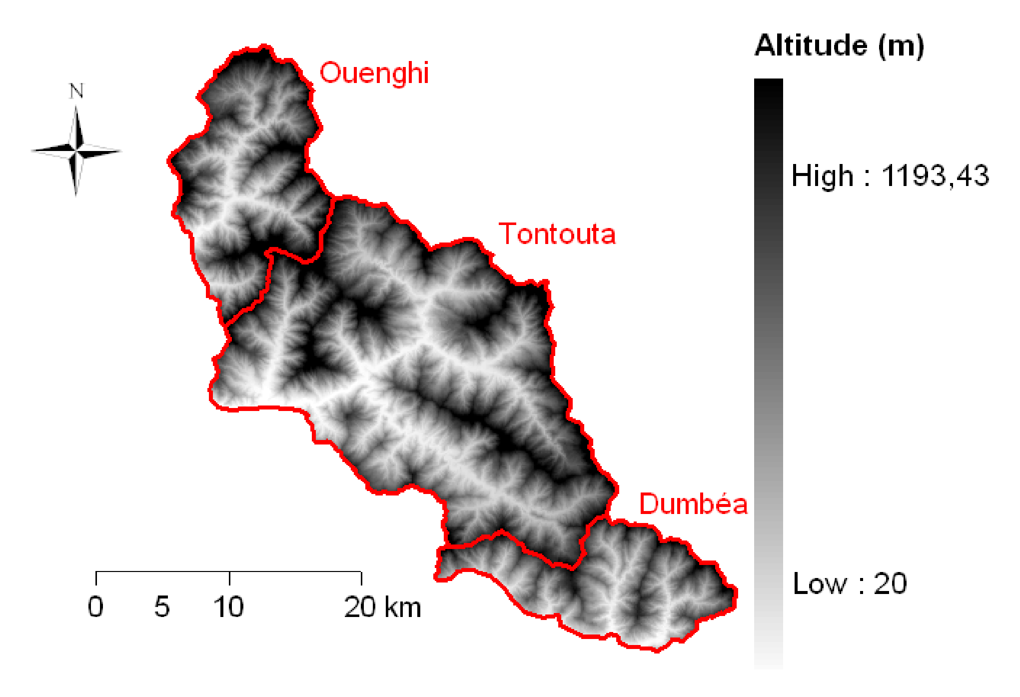


FIGURE 5.1 – Représentation de l'altitude pour les trois bassins versants de la zone d'étude.

5.2 Scénario de découverte de connaissances

Sur l'ensemble des trois bassins, seuls 3% du sol est érodé, le reste étant considéré comme non-érodé. Nous sommes typiquement face à des données aux classes disproportionnées. Dans cette section, nous développons un scénario de découverte de connaissances basée sur l'extraction de règles de caractérisation OVE afin de répondre aux attentes des experts géologues. Le schéma décrivant notre scénario est reporté dans la figure 5.2. Plus précisément, nous allons :

- extraire des règles de caractérisation OVE . Cela nous permet de caractériser les phénomènes d'érosion des sols couverts par notre ensemble de règles et ainsi d'identifier les combinaisons de facteurs des couches thématiques propices à l'érosion.
- construire un modèle prédictif basé sur l'ensemble des règles extraites pour le phénomène d'érosion et le valider.
- proposer une méthode d'estimation de l'aléa érosion qui découle directement de notre modèle prédictif.

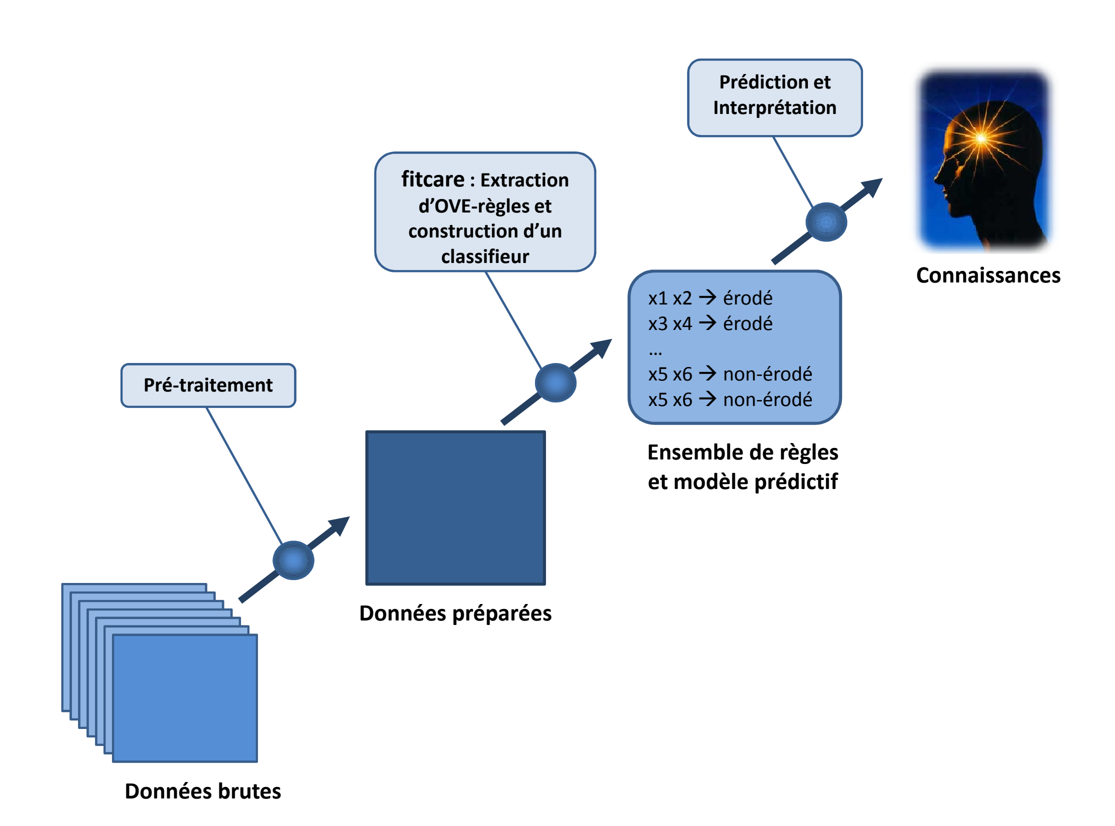


FIGURE 5.2 – Scénario d'extraction de connaissances dans les données d'érosion en Nouvelle-Calédonie

5.2.1 Pré-traitement

L'hétérogénéité des données fait qu'elles ne sont pas exploitables directement. En effet, les différentes couches de données ne sont pas à la même granularité. Pour pallier ce problème, nous ramenons toutes les couches de données à l'échelle la plus grande (i.e. celle de la couche d'altitude issue du modèle numérique de terrain à 30m). Puis chaque couche est transformée en grille dont chaque cellule (pixel) représente une partie du sol de $30m \times 30m$. Chacun de ces pixels est décrit par six attributs et étiqueté par “sol érodé” ou “ sol non-érodé”. Pour passer de la forme de grille à une base de données transactionnelles, chaque pixel devient une transaction (objet). Pour l'ensemble des trois bassins versants, cette transformation des rasters de données vers les données transactionnelles aura généré environ 8.10^5 objets.
Lors de la prise des photos satellitaires, une petite partie des sols (correspondant à environ 10000 pixels) était couverte par des nuages, rendant le calcul de l'indice de brillance impossible. Ces pixels sont éliminés des données.
Après une première étude dans [GRM$^+$07], nous avons remarqué que la couche thématique sur la pente biaisait le problème de la caractérisation des sols et les modèles prédictifs construits. En effet, il en ressort que les sols de faible pente sont quasiment toujours associés à des phénomènes d'érosion – ce qui est contre-intuitif. En fait, dans les zones étudiées, la très grande majorité des sols de pente faible correspondent aux lits de rivières. Suivant le conseil des experts du domaine, pour écarter ces zones de notre étude, nous appliquons un filtre afin de ne retenir que les données où la pente est supérieure à 15°. Notons toutefois, que ce filtre grossier n'écarte qu'une infime partie des plateformes minières et des rares plateaux présents dans la zone d'étude en raison de la précision à 30m.
Pour la tâche de prédiction, avec les conseils des experts, nous choisissons le bassin de la Tontouta comme base d'apprentissage. Ce bassin est le plus grand et présente la plus grande diversité en terme de phénomènes d'érosion. Les données de ce bassin sont discrétisées selon la méthode de Fayyad et Irani [FI93]. Le schéma de discrétisation obtenu est reporté sur les données des bassins de la Ouenghi et de Dumbéa qui serviront de validation.
Nous disposons maintenant de données préparées à l'extraction de règles de caractérisation `OVE` .

5.2.2 Extraction des règles de caractérisation `OVE`

La base de données sur le bassin de la Tontouta constitue notre base d'apprentissage des règles de caractérisation `OVE` . Le nombre de pixels étant conséquent (environ 4.9×10^5 pixels), nous utilisons la moitié des données (en respectant la répartition originale des classes) pour évaluer chaque matrice de paramètres produites. De cette manière, le processus d'optimisation par hill-climbing est plus rapide. L'exécution de cette étape se fait par la seule commande

```
fitcare -c "sol_érodé sol_non-érodé" -t 0.5 tontouta.txt
```

et dure plusieurs heures. L'ensemble $S_{\texttt{OVE}}$ des 66 règles de caractérisation OVE en sortie est reporté dans la table 5.1 pour les 32 règles concluant sur la classe "sol érodé" et dans la table 5.2 pour celles concluant sur la classe "sol non-érodé". Les paramètres optimaux découverts par `fitcare` sont reportés dans la matrice suivante :

$$\Gamma_{best} = \begin{pmatrix} \gamma_{erode,erode} & \gamma_{erode,non-erode} \\ \gamma_{non-erode,erode} & \gamma_{non-erode,non-erode} \end{pmatrix} = \begin{pmatrix} 0,03998 & 0,03969 \\ 0,03998 & 0,41428 \end{pmatrix}$$

Le taux de couverture de la base d'entraînement par $S_{\texttt{OVE}}$ est de 90,7% (respectivement 93%) pour les objets-pixels de classe "sol érodé" (respectivement "sol non érodé") – ce qui nous donne une bonne couverture de la base. Le taux d'accroissement global de $S_{\texttt{OVE}}$ est de 1,95 (voir définition au chapitre 4). La matrice Γ_{best} obtenue est en fait la matrice obtenue lors de la phase d'initialisation de `fitcare` . La phase d'optimisation ne permet pas d'obtenir de meilleurs paramètres tout en maintenant les taux de couverture (maximaux) obtenus lors de la phase d'initialisation. Pour chaque règle nous reportons aussi la fréquence de leur corps dans chacune des classes et la valeur du taux d'accroissement.

Analyse de l'ensemble des règles OVE

Dans la table 5.1, nous remarquons la règle $solnu \rightarrow solerode$ très fréquente (17%) et avec une forte valeur de TA (30). Cette règle confirme l'intuition des experts du domaine : un sol dénudé (qui est un type d'occupation du sol) sera plus propice à l'apparition du phénomène d'érosion. Il en est de même pour les autres règles à fort taux d'accroissement (en gras dans la table 5.1 lorsque $TA > 4$). Les types de sols "alluvions actuelles et récentes", "décharges minières non-contrôlées", "zones d'exploitations et déblais miniers" sont aussi propices à l'érosion. D'autre part, nous remarquons que certains facteurs reviennent souvent dans le corps des règles : le type de sol "latérites indifférenciées sur péridotite" et l'occupation du sol (végétation) "maquis minier clairsemé". Ce sont aussi des facteurs (combinés à d'autres) propices à l'érosion. Outre toutes ces confirmations pour les experts, l'ensemble de règles OVE permet de quantifier les phénomènes d'érosion observés grâce aux fréquences des corps de règles dans les deux classes. De plus, le taux d'accroissement donne une valeur qualitative aux phénomènes observés.

En ce qui concerne la caractérisation des zones moins propices à l'érosion, les règles du type $X \rightarrow sol_non_eorde$ sont reportées dans la table 5.2. Comme attendu par les experts, le type de sol "harzburgites" et le type de végétation "maquis minier dense" sont prédominants dans le corps des règles OVE et sont des facteurs limitant l'apparition des phénomènes d'érosion. De même, les forêts denses assurent une bonne tenue du sol.
Toutefois, trois règles à forte valeur de TA son plutôt inattendues :

$2483 < precipitations < 2872 \rightarrow sol_non_erode$

$2872 < precipitations < 3008 \rightarrow sol_non_erode$

Itemsets X (corps de règles)	$freq_r(X, r_{erode})$	$freq_r(X, r_{non-erode})$	$TA(X)$
harzburgites 20,9<pente<27,3	0.0176887	0.0134585	1.3143
harzburgites maquis_minier_clairsemé 1481<precipitations<1826	0.0197851	0.0134222	1.4741
20,9<pente<27,3 1481<precipitations<1826	0.0174921	0.00485659	3.6017
maquis_minier_clairsemé 32,3<pente<40,3	0.0210954	0.013913	1.5162
maquis_minier_clairsemé 40,3<pente<48,1	0.0212264	0.014511	1.4628
maquis_minier_clairsemé pente>48,1 132<altitude<442	0.0186714	0.0110405	1.6912
maquis_minier_clairsemé pente>48,1 1481<precipitations<1826.	0.024109	0.0108003	2.2323
maquis_minier_clairsemé 541<altitude<728	0.0346567	0.0163534	2.1192
maquis_minier_clairsemé latérites_indifférenciées_sur_péridotite	0.0447458	0.015688	2.8522
maquis_minier_clairsemé 1956<precipitations<2197	0.0259434	0.0122607	2.116
32,3<pente<40,3 541<altitude<728	0.0170335	0.0105082	1.621
32,3<pente<40,3 latérites_indifférenciées_sur_péridotite	0.026533	0.00751302	3.5316
32,3<pente<40,3 1481<precipitations<1826	0.0246331	0.0113706	2.1664
40,3<pente<48,1 latérites_indifférenciées_sur_péridotite	0.0233884	0.00782931	2.9873
40,3<pente<48,1 1481<precipitations<1826	0.023978	0.0143969	1.6655
17,5<pente<20,9	0.0281053	0.00822509	3.417
pente>48,1 541<altitude<728 1481<precipitations<1826	0.0214885	0.014091	1.525
17,5<pente	0.0235849	0.00498448	**4.7317**
728<altitude<851 latérites_indifférenciées_sur_péridotite	0.0227987	0.0104373	2.1843
728<altitude<851 1481<precipitations<1826	0.0271226	0.0102438	2.6477
541<altitude<728 latérites_indifférenciées_sur_péridotite	0.0604036	0.0116368	**5.1907**
541<altitude<728 1956<precipitations<2197	0.0303328	0.0156966	1.9324
sol_nu	0.178656	0.00592296	**30.1633**
442<altitude<541 latérites_indifférenciées_sur_péridotite	0.0184093	0.00451265	**4.0795**
442<altitude<541 1956<precipitations<2197	0.0165094	0.0102282	1.6141
132<altitude<442 latérites_indifférenciées_sur_péridotite	0.0275157	0.00637233	**4.318**
latérites_indifférenciées_sur_péridotite 1481<precipitations<1826	0.0678066	0.0127931	**5.3002**
alluvions_actuelles_et_récentes_Fyz	0.0193265	0.00158487	**12.1944**
décharges_minières_non_contrôlées_et_coulées_de_matériaux	0.0220781	0.00134636	**16.3984**
zones_d'exploitations_et_déblais_miniers	0.0280398	0.000888358	**31.5636**
78<altitude<132	0.0229298	0.0154841	1.4809
1255<precipitations<1299	0.0239125	0.00654689	3.6525

Table 5.1 – Règles de caractérisation OVE concluant sur la classe “sol érodé”.

$precipitations > 3008 \rightarrow sol_non_erode$
que nous pouvons regrouper et traduire par :
$fortes_precipitations \rightarrow sol_non_erode$
A priori, cela semble contre-intuitif car de fortes précipitations sur une zone seraient un facteur d'apparition des phénomènes d'érosion. Nous pensons que cela est dû à la nature même de la couche de données sur les précipitations. La granularité à $30m$ n'implique pas que nous disposons de données de pluviométrie pour chaque zone de $30m \times 30m$. En effet, seuls quelques pluviomètres sont disponibles par bassin versant et les données récoltées sont extrapolées en utilisant l'altitude des zones aux alentours via le modèle `AURELHY` sur le reste de la zone d'étude. Intuitivement, plus l'altitude est élevée, plus la quantité de précipitations sera grande. Dans les zones étudiées, les sols en haute altitude sont "souvent" des sols non-propices à l'érosion (e.g. harzburgites). C'est pourquoi nous obtenons ce type de règle. Toutefois, pour confirmer notre hypothèse, une étude plus poussée focalisée sur ces zones est nécessaire.

Itemsets X (corps de règles)	$freq_r(X, r_{erode})$	$freq_r(X, r_{non-erode})$	$TA(X)$
2483<precipitations<2872	0.00817717	0.0522101	6.3849
harzburgites 851<altitude<1193	0.00889676	0.0360154	**4.0481**
harzburgites maquis_minier_dense 1481<precipitations<1826	0.0146535	0.0417467	2.8489
harzburgites pente>48,1 541<altitude<728	0.0164198	0.044539	2.7125
harzburgites pente>48,1 442<altitude<541	0.0117097	0.028579	2.4406
harzburgites pente>48,1 132<altitude<442 1481<precipitations<1826	0.0117751	0.0240419	2.0418
harzburgites pente>48,1 1956<precipitations<2197	0.0155693	0.0278662	1.7898
harzburgites 728<altitude<851	0.0111864	0.032229	2.8811
harzburgites 1826<precipitations<1956	0.0151768	0.0305826	2.0151
851<altitude<1193 maquis_minier_clairsemé	0.0162235	0.0189543	1.1683
851<altitude<1193 maquis_minier_dense	0.00346712	0.0193219	**5.5729**
851<altitude<1193 pente>48,1	0.00935468	0.0323688	3.4602
forêt_dense	0.0113826	0.090274	**7.9309**
32,3<pente<40,3 maquis_minier_dense	0.00771925	0.0207819	2.6922
32,3<pente<40,3 132<altitude<442	0.0164198	0.0170939	1.0411
40,3<pente<48,1 maquis_minier_dense	0.00765383	0.0264649	3.4577
40,3<pente<48,1 132<altitude<442	0.0135414	0.0205385	1.5167
maquis_minier_dense pente>48,1 132<altitude<442	0.00902759	0.0497785	**5.514**
maquis_minier_dense pente>48,1 1481<precipitations<1826	0.012691	0.0325604	2.5656
maquis_minier_dense 728<altitude<851	0.0064109	0.0190751	2.9754
maquis_minier_dense 541<altitude<728	0.0139993	0.0394911	2.8209
maquis_minier_dense 442<altitude<541	0.00614923	0.0240333	3.9083
maquis_minier_dense 132<altitude<442 1481<precipitations<1826	0.00856967	0.0258453	3.0159
maquis_minier_dense latérites_indifférenciées_sur_péridotite	0.0123639	0.0176116	1.4244
maquis_minier_dense 1956<precipitations<2197	0.010663	0.0233671	2.1914
maquis_minier_dense 1310<precipitations<1481	0.00667257	0.01737	2.6032
latérites_minces_sur_péridotite	0.0118405	0.0234879	1.9837
pente>48,1 1826<precipitations<1956	0.00778466	0.0246942	3.1722
savane_arbustive_et_arborée	0.0109247	0.0375548	3.4376
2872<precipitations<3008	0.00098126	0.0200467	**20.4295**
precipitations>3008	0.000130835	0.0216879	**165.7653**
132<altitude<442 1956<precipitations<2197	0.0118405	0.0189146	1.5974
132<altitude<442 1310<precipitations<1481	0.0125601	0.0170473	1.3573
2289<precipitations<2435	0.0129526	0.0238262	1.8395

TABLE 5.2 – Règles de caractérisation `OVE` concluant sur la classe "sol non-érodé".

5.2.3 Construction d'un modèle prédictif

`fitcarc`, l'algorithme de classification supervisée associé à `fitcare` nous permet de construire un classifieur à partir des règles de $S_{\texttt{OVE}}$. Nos données test sont les bassins de la Ouenghi et de la Dumbéa. La phase de prédiction de l'érosion des sols dans ces zones se fait via les commandes :

```
fitcarc -s -r tontouta_ove_rules.txt dumbea.txt
fitcarc -s -r tontouta_ove_rules.txt ouenghi.txt
```

Nous reportons les résultats de précision pour chaque bassin dans les matrices de confusion des figures 5.3 et 5.4.

Dumbéa Prédictions	Classes réelles	
	sol non-érodé	sol érodé
sol non-érodé	112827	743
sol érodé	14437	**926**

FIGURE 5.3 – Matrice confusion pour les résultats de précision sur le bassin de la Dumbéa.

Ouenghi Prédictions	Classes réelles	
	sol non-érodé	sol érodé
sol non-érodé	137016	835
sol érodé	31173	**3194**

FIGURE 5.4 – Matrice confusion pour les résultats de précision sur le bassin de la Ouenghi.

La classe mineure "sol érodé" est la classe qui nous intéresse. Les taux de vrais positifs obtenus par `fitcare` sont respectivement de 55% et 79% pour les bassins de la Dumbéa et de la Ouenghi. Bien que les résultats pour le bassin de la Dumbéa soient moyens, il est bon de noter que dans les mêmes conditions d'expérimentations, `NB` (respectivement `C4.5`) obtient des taux de vrais positifs de 17% et 33% (respectivement 2% et 18%) pour les bassins de la Dumbéa et de la Ouenghi. Nous remarquons aussi les taux relativement bas de faux positifs : 11% pour la Dumbéa et 18% pour la Ouenghi. Toutefois, les rapports de vrais positifs sur faux positifs – $926/(926+14437) \simeq 0,06$ pour la Dumbéa et $3194/(3194+31173) \simeq 0,09$ – indiquent que le classifieur présente un biais vers la classe minoritaire "sol érodé". Ce biais est cependant léger si l'on considère les faibles taux de faux positifs : $14437/(14437 + 112827) \simeq 0,11$ pour la Dumbéa et $3194/(3194 + 31173) \simeq 0,18$ pour la Ouenghi.

Le classifieur ainsi construit nous permet de générer une cartographie "prédite" des zones d'érosion dans les deux bassins test. Ces cartes sont reportées dans les figures 5.5 et 5.6.

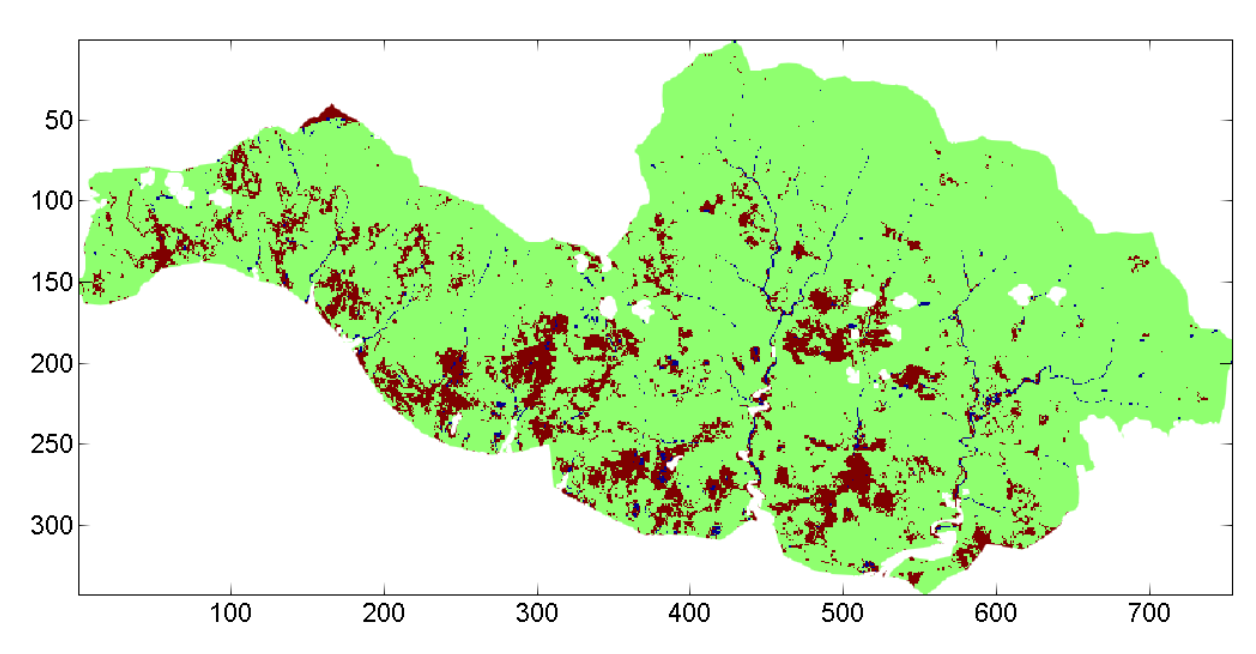


FIGURE 5.5 – Cartographie des zones d'érosion par prédiction avec `fitcare` sur le bassin de la Dumbéa.

Il apparaît visuellement sur les cartes des figures 5.5 et 5.6 que les zones érodées sont surestimées. En effet, 13% des sols du bassin de la Dumbéa et 24% des sols de la Ouenghi sont prédits "sol érodé" par notre classifieur, alors que l'indice de brillance (couche test) nous indique respectivement 3% et 5%. Une étude détaillée des zones concernées par ce biais montrent que ces zones sont "souvent" des sols de type "latérites indifférenciées sur péridodite". Ce type de sol est souvent présent dans le corps de règles concluant sur "sol érodé" et ayant un taux d'accroissement moyen (entre 1 et 3,5) et explique une grande partie des faux positifs.

5.2.4 Estimation de l'aléa érosion

Pour chaque objet t, au lieu de la version simplifiée de la prédiction ("sol érodé" ou "sol non-érodé"), `fitcarc` a la possibilité de fournir les fréquences normalisées (dans chaque classe) des règles de $S_{\texttt{OVE}}$ qui couvrent t. Ainsi, au lieu de ne garder que la classe c_i qui maximise le score de ressemblance de t avec c_i, on normalise les scores obtenus de manière à obtenir une somme de ces scores égale à 1 pour t. Plus la fréquence normalisée liée à la classe "sol érodé" est proche de 1 plus l'objet concerné a de chances d'être érodé. Nous utilisons cette valeur pour donner une estimation de l'aléa érosion dans les bassins versants de la Ouenghi et de la Dumbéa. Le calcul des fréquences normalisées est réalisé par les commandes suivantes :

```
fitcarc -r tontouta_ove_rules.txt dumbea.txt
```

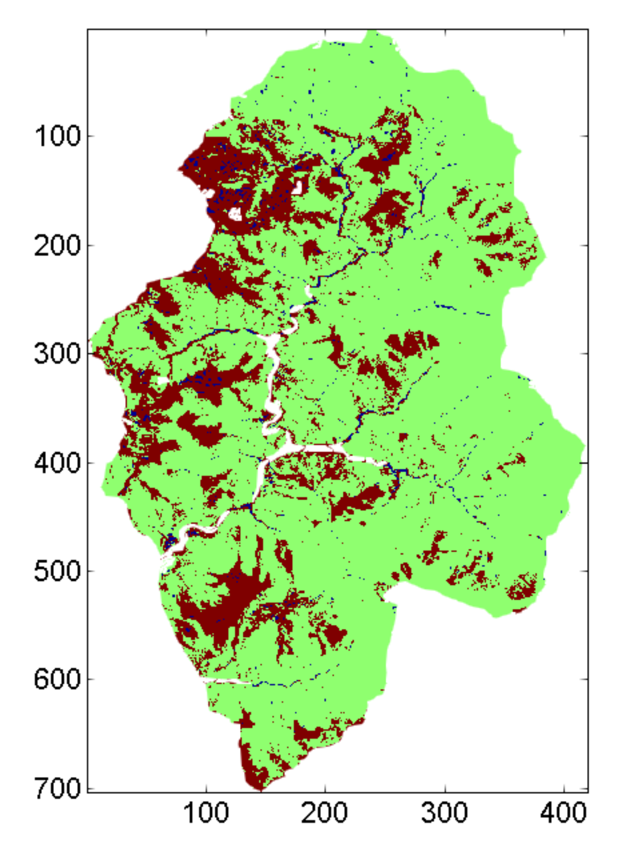


FIGURE 5.6 – Cartographie des zones d'érosion par prédiction avec `fitcare` sur le bassin de la Ouenghi.

```
fitcarc -r tontouta_ove_rules.txt ouenghi.txt
```

Dans les figures 5.7 et 5.8, nous reportons les cartes de l'aléa érosion pour les deux bassins versants test.

5.3 Discussion

Le déroulement du processus de découverte de connaissances présenté dans ce chapitre a permis de produire un ensemble de règles de caractérisation du phénomène d'érosion des sols en Nouvelle-Calédonie. Nombre de ces règles ont été confirmées et validées par les experts du domaine d'application. L'ensemble de règles extrait permet aussi de quantifier et qualifier les phénomènes observés. De plus, le modèle prédictif à base de règles issu de ce processus offre la possibilité de générer automatiquement une cartographie des zones d'érosion et de l'aléa érosion.

Cependant, la cartographie automatique générée par `fitcare` subit l'effet dit de "poivre et sel", c'est-à-dire, il peut arriver que des zones identifiées comme érodées soient

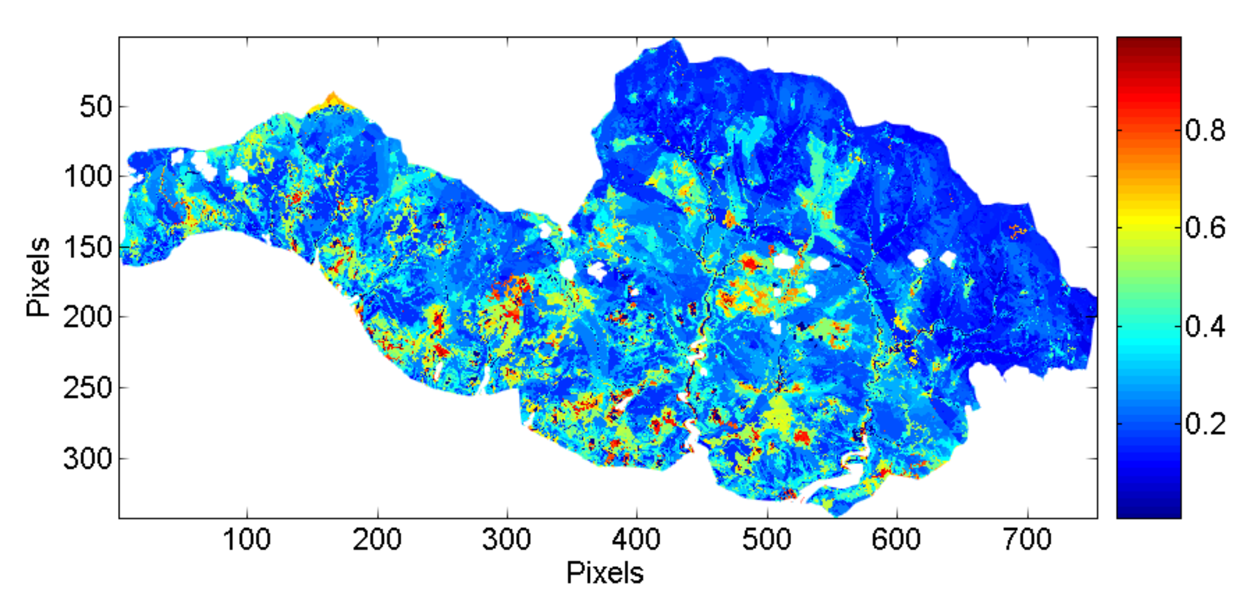


FIGURE 5.7 – Estimation de l'aléa érosion pour le bassin de la Dumbéa.

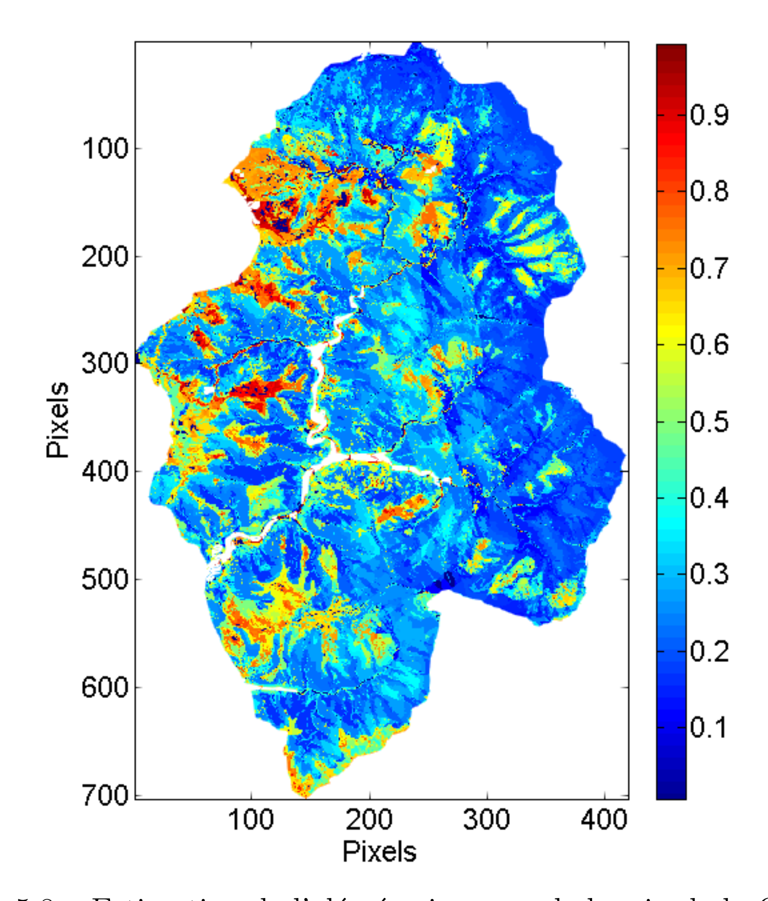


FIGURE 5.8 – Estimation de l'aléa érosion pour le bassin de la Ouenghi.

en fait des pixels isolés. La figure 5.9 donne un exemple de cet effet. Ces pixels n'ont a priori pas de signification intéressante selon les experts. Nous pensons que c'est principalement dû à deux raisons : *(i)* la couche des sols érodés/non-érodés est issue d'une segmentation et présentait déjà cet effet poivre et sel – typique des méthodes de segmentation. *(ii)* La nature même des données (i.e., des grilles de pixels) et donc la prédiction de classe pour des pixels sont aussi connues pour être sujettes à l'effet poivre et sel. Si une première solution peut consister en un post-traitement pour éliminer les pixels isolés, une autre solution serait de considérer des couches de données segmentées en zones plus significatives que de simples pixels. Ainsi, un objet de notre base ne serait plus un pixel mais une zone (groupement de plusieurs pixels) obtenue par segmentation.

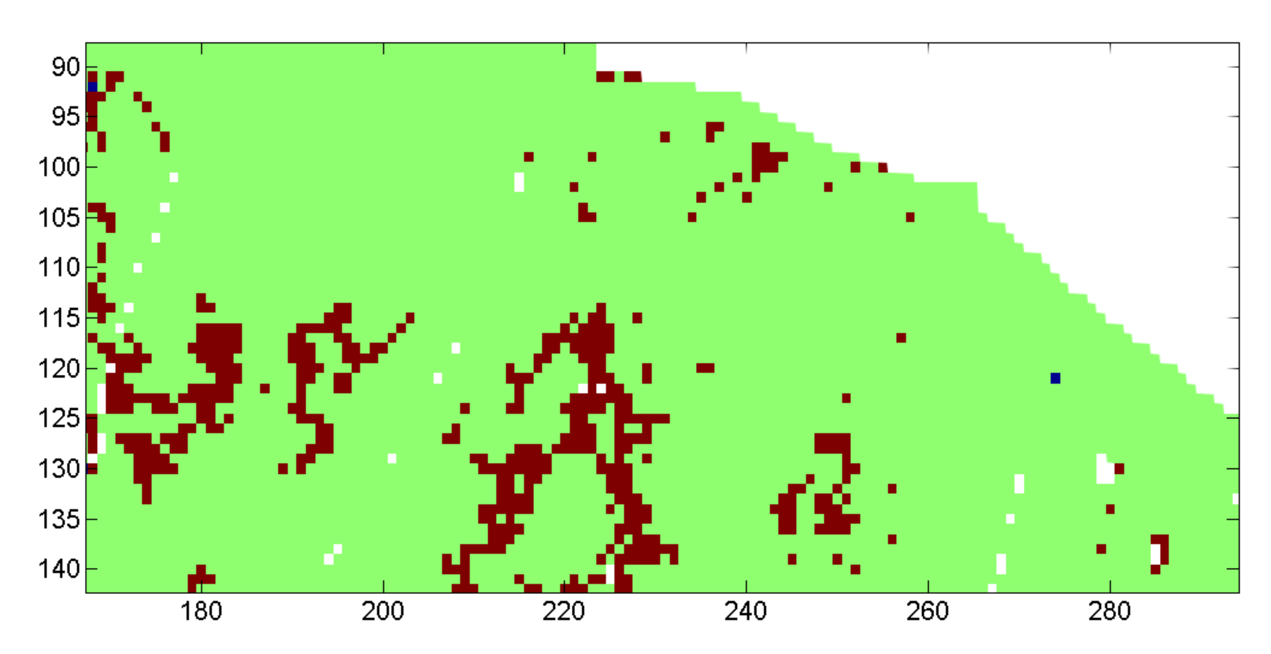


FIGURE 5.9 – Effet poivre et sel sur une partie zoomée du bassin de la Dumbéa.

Cinquième partie

Conclusion & Perspectives

Conclusion & Perspectives

Dans ce mémoire nous nous sommes intéressés au calcul de motifs sous contraintes et à leur utilisation pour des problèmes difficiles de classification supervisée. Nos deux principales contributions proposent des solutions à deux problèmes clairement identifiés : *(i)* la construction de descripteurs pour la classification supervisée dans les domaines aux attributs bruités ; *(ii)* la classification supervisée à base de motifs dans les bases de données multi-classes disproportionnées.

Construction de descripteurs basée sur les propriétés de fermeture

Avant notre travail, les techniques de construction de descripteurs pour la classification supervisée (ainsi que les techniques de classification associative) utilisant les propriétés de fermeture exploitaient tantôt les itemsets 0-libres tantôt les itemsets fermés pour caractériser les classes. S'il existait des divergences d'avis à ce sujet, notre étude a pointé du doigt les raisons de cette divergence : *(i)* lorsqu'on procède à une extraction dans la base entière sans tenir compte de l'attribut classe, les itemsets 0-libres sont choisis ; *(ii)* lorsqu'on procède à une extraction par classe, les itemsets fermés sont choisis. Dans les deux cas, le choix est fait par souci de redondance. Le dilemme est ainsi lié à la méthode d'extraction. De plus, pour obtenir des motifs intéressants pour la caractérisation des classes, un post-traitement est nécessaire.

Dans ce manuscrit, nous proposons une méthode de construction de descripteurs basés sur des motifs qu'on extrait en tenant compte de l'attribut classe. En extrayant les itemsets γ-fréquents δ-libres dont la δ-fermeture contient un attribut classe, nous traitons à la fois la redondance dans l'ensemble des itemsets fréquents et le pouvoir discriminant des règles δ-fortes de caractérisation construites sur ces motifs. Notre approche `FC` est ainsi sans post-traitement. Les itemsets δ-libres ayant déjà servi dans la construction de motifs tolérants aux erreurs comme dans la caractérisation de groupements dans les contextes bruités, nous montrons dans notre approche qu'un ajustement stratégique des valeurs de δ (par rapport à γ) permet de construire des descripteurs robustes et tolérants au bruit

d'attribut. Ainsi, des classifieurs classiques comme C4.5 , NB , SVM voient leur précision améliorée sur des données munies des nouveaux descripteurs que nous construisons, et ce pour des données originales comme pour des données dont les attributs sont bruités.

Perspectives

Les itemsets δ-libres auront bientôt dix années d'applications fructueuses depuis leur création. Si l'application directe de notre approche FC n'est pas adaptée aux données dont les classes sont bruitées, nous n'avons pas étudié leur utilisation dans des processus de correction de bruit de classe ou de suppression d'objets dont la classe serait bruitée – ce qui serait une première piste pour des travaux futurs. D'autre part, leur définition originelle oblige une unique valeur de δ. Une autre piste de travail serait d'étudier différentes valeurs de δ (dans la définition de la δ-liberté et de δ-fermeture) pour les attributs classes et les autres attributs.

Bases de données multi-classes disproportionnées

Les problèmes multi-classes disproportionnées disposaient de deux familles de solutions en classification à base de motifs : *(i)* les approches OVA tentent de caractériser chaque classe par rapport au reste des données ; *(ii)* les approches OVO divisent les problèmes à n classes en plusieurs sous-problèmes à deux classes, les classifieurs obtenus sur les sous-problèmes sont ensuite mixés par divers procédés. Si les approches OVA peuvent mener à des incohérences (conflits de règles, biais vers la classe majoritaire, ...) dans ce type de problème, les approches OVO travaillent en post-traitement des différents classifieurs issus des sous-problèmes afin de traiter les redondances et conflits de règles. De plus, en classification associative OVA comme OVO , il est nécessaire de fixer un ou souvent plusieurs paramètres qui influent grandement sur la qualité du classifieur qui en découle.

Dans ce manuscrit, nous avons décidé de développer une approche hybride qu'on pourrait situer entre approches OVA et OVO . Notre approche de classification associative OVE permet de caractériser chaque classe du problème par rapport à chaque autre classe. Pour ce faire, il est nécessaire de considérer n^2 paramètres (dont n seuils de fréquence minimum et n^2-n seuils d'infréquence ou d'erreurs maximum) qu'on peut voir sous forme de matrice de paramètres Γ. Pour répondre au problème posé et garder une certaine cohérence dans l'ensemble de règles respectant les paramètres seuils, nous avons imposé des contraintes entre seuils. La formalisation de notre approche avec une telle matrice de paramètres et ses contraintes a vu naître un nouveau type de règle : les règles de caractérisation OVE (OVE-CRs). Les OVE-CRs apportent une solution à chacun des problèmes énoncés qui a motivé sa création (conflits de règles, redondance, biais, post-traitement). Les corps des OVE-CRs peuvent aussi être vus comme une généralisation de la notion d'itemset émergent

adaptée aux données multi-classes disproportionnées.
Si l'extraction des OVE-CRs en fonction de Γ n'est pas la tâche la plus difficile, trouver les bons paramètres est problématique – sachant que ces paramètres sont différents selon les données. Dans notre approche, nous faisons face à la multitude de paramètres (inhérent à notre formalisation du problème) par une technique automatique et intelligente de paramétrage. Notre prototype `fitcare` intègre ainsi matrice de paramètres, contraintes, paramétrage automatique et extraction des OVE-CRs pour construire un classifieur libre de tout paramétrage dédié aux données multi-classes disproportionnées. L'originalité de notre approche vient compléter les résultats probants en terme précision sur les classes mineures.

Perspectives

Les perspectives de travail à la suite de notre approche OVE `fitcare` sont multiples. Tout d'abord, à court terme, on pourrait penser à expérimenter notre méthode dans des contextes bruités – et ainsi évaluer sa robustesse face aux données imparfaites. D'autre part, `fitcare` est basé sur une technique *simple* d'optimisation pour la recherche locale de solutions : le hill-climbing. Il existe beaucoup d'autres algorithmes de recherche locale. Une étude plus approfondie des différentes méthodes et de leur application à notre problème pourrait nous mener à de meilleures solutions pour Γ à la fois en terme d'efficacité de calcul, de pertinence et de précision.

A moyen terme, une piste de travail pourrait être d'adapter notre approche `fitcare` aux problèmes de classification supervisée sensible aux coûts de classification. Dans ce type de problème, les erreurs de classification d'une classe c_i à une classe c_j sont différentes selon la classe étudiée et la classe où les erreurs sont faites. Classes disproportionnées et coûts de classification sont intimement liées [CCHJ08]. Une matrice de coûts de classification Γ_{cost} de la même taille que Γ dirigerait alors le processus d'optimisation afin de limiter les coût des erreurs de classification. Ainsi, couverture, intérêt des règles et coûts de classification seraient les critères à optimiser. Clairement, cela augmente la difficulté du problème.

Caractérisation de l'érosion des sols en Nouvelle-Calédonie

Avant notre étude, la cartographie des zones d'érosion ainsi que l'estimation de l'aléa érosion à l'échelle de région étaient coûteuses en temps et en argent. Bien que l'érosion ait un fort impact sur l'environnement, les zones érodées dans les régions étudiées ne couvrent qu'environ 3% des sols. La mise en commun de données sur différents paramètres physiques considérés par les experts comme facteurs du phénomène d'érosion résulte en une grosse base de données dont la répartition des classes "sol érodé/sol non-érodé" est très déséquilibrée.

Dans ce manuscrit, afin de répondre aux attentes des experts, nous avons développé un processus complet de découverte de connaissances dont l'étape clé est l'extraction de règles de caractérisation `OVE` . Une première analyse des règles extraites a permis de confirmer certaines combinaisons de facteurs responsables du phénomène d'érosion. Mais aussi, grâce à des mesures d'intérêt des règles `OVE` intuitives pour les experts (e.g. la fréquence et le taux d'accroissement), il nous est maintenant possible de qualifier les facteurs d'érosion. De plus, l'ensemble des règles `OVE` sert comme base pour la construction d'un classifieur dédié aux données aux classes disproportionnées. Nous avons alors proposé un modèle prédictif performant pour les zones d'érosion ainsi qu'une méthode d'estimation de l'aléa érosion en Nouvelle-Calédonie. La cartographie des zones d'érosion et de l'aléa érosion s'en trouve facilitée.

Perspectives

Une perspective intéressante pour les travaux de recherche sur la caractérisation de l'érosion des sols serait de prendre en compte la dimension spatiale dans les données. En effet, par exemple, la proximité de certains sols (de type latérite, ou à forte pente ou encore couvert par du maquis minier clairsemé) et des plateformes minières est intuitivement une combinaison propice à l'apparition de l'érosion. Cette combinaison est inaccessible via nos méthodes qui ne prennent pas en compte la notion de voisinage des zones.
Depuis 2009, l'équipe “Data Mining” du PPME a étendu ses axes de recherche à la fouille de données spatiales (Spatial Data Mining). Les avancées dans ce nouvel axe offre de belles perspectives dans ce domaine d'application. A titre d'exemple, dans [SFFG^{+}09, FSFG^{+}10], une première approche d'extraction de motifs spatiaux (i.e., co-locations intéressantes d'événements) est proposée pour la caractérisation de l'érosion des sols.

Appendice - Description des données d'expérimentation

Base de données `meningite`

La base de données `meningite` répertorie des informations sur 328 enfants atteints de méningite aiguë. Il existe deux types de méningite : d'origine virale (la majorité des cas, soit 245) ou d'origine bactérienne (un quart des cas, soit 84). Les 23 attributs discrets de descriptions des différents sont ou des paramètres épidémiologiques (e.g. saison, âge, sexe) ou des paramètres sur l'état du patient (e.g. température, forme) encore des résultats d'analyse (e.g. glucose).
Les traitements indiqués contre la méningite diffèrent fortement selon le type. Cela va de la simple surveillance médicale pour le virus à la prescriptions d'anti-biotiques appropriés contre la bactérie. C'est pourquoi, il est important de rapidement, bien diagnostiquer.

Bases de données `UCI`

Dans la table 5.3, nous reportons une description succincte des données utilisées (nombre d'objets, d'attributs, de classes et la répartition des objets dans les différentes classes). Il faut savoir que le nombre d'attributs indiqué correspond au nombre d'attributs des données originales. Après discrétisation, le nombre d'attributs est souvent plus grand. Pour plus d'informations sur chacune de ces bases, nous invitons le lecteur à consulter le répertoire en ligne de bases de données de l'Université de Californie à Irvine [AN07] (http://archive.ics.uci.edu/ml/).

Données	#Objets	#Attributs	#Classes et répartition
balance-scale	625	4	288/49/288
breast-cancer	286	9	201/85
breast-w	699	9	458/241
car	1728	6	1210/384/69/65
colic	368	22	232/136
credit-a	690	15	307/383
diabetes	768	8	500/268
heart-c	303	13	165/138
heart-h	294	13	188/106
heart-s	270	13	150/120
hepatitis	155	19	32/123
iris	150	4	50/50/50
labor	57	16	20/37
sonar	208	60	97/111
ticTacToe	958	9	626/332
vote	435	16	267/168
waveform	5000	40	1657/1647/1696
wine	178	13	59/71/48
zoo	101	17	41/20/5/13/4/8/10

TABLE 5.3 – Description des bases de données UCI.

Appendice - Preuves

Dans cette section, nous reportons les preuves des différentes propositions utilisées au cours du manuscrit.

Preuve de la proposition 1

Afin d'alléger les notations, consider la table de contigence pour la règle $\pi : X \rightarrow c_i$ (voir figure 5.10).

FIGURE 5.10 – Table de contingence pour la règle $X \rightarrow c_i$ concluant sur un attribut classe c_i.

$X \rightarrow c$	c	$\bar{c}$	Σ
X	a	b	$a+b$
$\bar{X}$	c	d	$c+d$
Σ	$a+c$	$b+d$	$\lvert r\rvert = a+b+c+d$

Selon la définition du *facteur d'intérêt* [TSK05], X est dit positivement corrélé avec la classe c_i si

$$\widehat{corr}(\pi, c_i) = \frac{a \cdot |r|}{(a+b) \cdot (a+c)} > 1$$

$$\text{i.e., } \frac{|r|}{a+b+c+\frac{b \cdot c}{a}} > 1$$

D'autre part, soit un entier $\rho > 1$, X est un ρ-EP si

$$Gr(\pi, r_{c_i}) = \frac{a \cdot (b+d)}{b \cdot (a+c)} \geq \rho \text{ i.e., } a \cdot b + a \cdot d \geq \rho \cdot (a \cdot b + b \cdot c)$$

$$ab + ad \geq \rho ab + \rho bc \text{ alors } b + d \geq \rho b + \rho \frac{bc}{a}$$

$$d \geq (\rho-1)b + \rho\frac{bc}{a} > \frac{bc}{a} \text{ car } \rho > 1$$

$$\text{comme } d > \frac{bc}{a} \text{donc } \widehat{corr}(\pi, c_i) = \frac{|r|}{a+b+c+\frac{b \cdot c}{a}} > \frac{|r|}{a+b+c+d} = 1$$

Donc X est positivement corrélé avec c_i. □

Preuve de la proposition 2

Soient trois entiers γ, δ et $\rho > 1$ (respectivement des seuils pour la fréquence, le nombre d'erreur et le taux d'accroissement), nous savons que pour assurer qu'un corps X d'une δ-SCR $\pi : X \rightarrow c_i$ soit ρ-EP, il est suffisant que $\frac{\gamma-\delta}{\delta} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}|} \geq \rho$. C'est-à-dire, il est suffisant que

$$\frac{\gamma-\delta}{\delta} \geq \frac{|r_{c_i}|}{|r \setminus r_{c_i}|} \cdot \gamma \text{ donc } \frac{\gamma}{\delta} \geq \frac{|r_{c_i}|}{|r \setminus r_{c_i}|} \cdot \rho + 1$$

$$\text{i.e., } \frac{\gamma}{\delta} \geq \frac{\rho \cdot |r_{c_i}| + |r \setminus r_{c_i}|}{r \setminus r_{c_i}} \text{ donc } \gamma \cdot \frac{|r \setminus r_{c_i}|}{\rho \cdot |r_{c_i}| + |r \setminus r_{c_i}|} \geq \delta$$

$$\text{Observons que } \frac{|r \setminus r_{c_i}|}{\rho \cdot |r_{c_i}| + |r \setminus r_{c_i}|} > \frac{\gamma}{\rho} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}| + |r \setminus r_{c_i}|}, \text{ donc } \frac{\gamma}{\rho} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}| + |r \setminus r_{c_i}|} > \delta$$

Pour prendre en compte le fait que la distribution des classes peut être inégale, il suffit que l'inéquation précédente soit vérifiée pour la classe majoritaire. Donc, $\frac{\gamma}{\rho} \cdot \frac{|r \setminus r_{c_j}|}{|r|} > \delta$ (où c_j est la classe majoritaire) est une condition suffisante pour que le corps X soit un ρ-EP. □

Preuve de la proposition 3

Preuve par l'absurde. Soient deux `OVE`-CRs de S, $\pi : X \rightarrow c_i$ et $\pi' : Y \rightarrow c_j$ telles que $Y \subseteq X$ et $j \neq i$ et qui respectent les contraintes de fréquence et d'infréquence de Γ et les contraintes $\mathbb{C}_{ligne}$ et $\mathbb{C}_{colonne}$.
Nous avons $freq_r(X, r_{c_i}) \geq \gamma_{i,i}$ et $\forall k \neq i freq_r(X, r_{c_k}) < \gamma_{i,k}$ en raison de $\mathbb{C}_{ligne}$. De même, $freq_r(Y, r_{c_j}) \geq \gamma_{j,j}$ et $\forall l \neq j freq_r(X, r_{c_l}) < \gamma_{j,l}$. En raison de $\mathbb{C}_{colonne}$, et du fait que $Y \subseteq X$, nous avons l'inéquation suivante :

$$\gamma_{i,i} \leq freq_r(X, r_{c_i}) \leq freq_r(Y, r_{c_i}) < \gamma_{j,i}$$

et donc $\gamma_{i,i} < \gamma_{j,i}$,
ce qui entre en contradiction avec l'hypothèse de la contrainte $\mathbb{C}_{colonne}$. □

Appendice - Manuel de `fitcare`

Pour informations, `fitcare` et `fitcarc` *version* 0.16.2 sont utilisées pour les expérimentations reportées dans ce manuscrit.

NAME

Fitcare Is The Class Association Rule Extractor

SYNOPSIS

fitcare [options] dataset
fitcare --help | --version

OVERVIEW

From a classified data set, fitcare computes the bodies of the rules concluding on the classes such that every rule is frequent in one class and not frequent in any of the other classes.

Either every frequency threshold is bound to a parameter set by the user or these parameters are automatically learned.

Then, fitcarc can apply these rules on unclassified data.

RETURN VALUES

fitcare returns values which can be used when called from a script. They are conformed to sysexit.h.
* 0 is returned when fitcare terminates successfully.
* 64 is returned when fitcare was called incorrectly.
* 65 is returned when input data is not properly formatted.
* 74 is returned when data could not be read or written on the disk.

GENERAL

* fitcare --help (or simply fitcare -h) displays a reminder of the different options that can be passed to fitcare.
* fitcare --version (or simply fitcare -V) displays version information.

OPTIONS

Any option passed to fitcare may either be specified on the command line or in an option file. If an option is present both in the option file and on the command line, the latter is used.

The option file can be specified through option --opt. When omitted, a file named as the input data file + ".opt" is supposed to be the option file related to the input data file. For example, if dataset is "dataset.txt" and "dataset.txt.opt" exists, it is supposed to be the related option file.

The options have the same name in the option file as on the command line. Only long names can be used though. Arguments passed to an option are separated from the name of the option by "=". For example, these lines may constitute an option file:

```
verbose = true
iis = ": ,"
ois = ","
```

INPUT DATA

The name of the file containing the input data set must be passed as argument dataset.

The syntax of the input data set is flexible. Every line must be either empty (it is ignored) or contain all the attributes of an object. Several different characters may be used to separate the attributes. The attributes can be any string (as far as they do not include any of the separators!). Let us take an example. This line could be part of an input data set:

```
male basketball, gymnastics : good
```

The object related to this line is described by four attributes: male, basketball, gymnastics and good. One of them should be a class attribute. The attributes are separated by the characters " " or/and "," or/and ":". The fact that two attributes may be separated by several separators does not raise any trouble.

To be properly understood, the user must indicate the separators otherwise defaults are used (" "). The related option is --iis for Input Item Separators. In the previous example, fitcare can be called as follows:

```
$ fitcare --iis " ,:" dataset.txt
```

FIXED THRESHOLDS

To extract class association rules respecting given frequency thresholds, one must provide them in the form of a file containing a matrix. Every (non-empty) line of this matrix starts with the class attribute. When extracting the class association rules concluding on this class, the following float numbers are the frequency thresholds. The first one correspond to the frequency threshold in the first class, the second one to the frequency threshold in the second class, etc.

For example, this could be the frequency matrix for the extraction of class association rules in a data set organized in three classes:

```
excellent: 0.5 0.2 0
good: 0.2 0.4 0.2
bad: 0.1 0.2 0.5
```

The characters that can be used to separate the classes and the thresholds from each other are ":", ",", " " and tabulation.

To avoid conflicts, every value on the diagonal (corresponding to the minimal frequency threshold in the class the extracted class association rules are concluding on) must be strictly greater than every other threshold on its column.

Option --mat (-m) is used to set the matrix file name. When omitted, the matrix file name is supposed to be the input file name + ".mat". For example, if dataset is "dataset.txt", the default output file name is "dataset.txt.mat". The matrix file is mandatory.

LEARNED THRESHOLDS

To extract class association rules with learned frequency thresholds, one must only provide the list of strings related to the class attributes in the input data.

This list is set with option --classes (-c). On the command line, do not forget to protect it by using double quotes ("). For example, the following command can be used to compute rules concluding on the class attributes "yes", "no" or "maybe":

```
$ fitcare -c "yes no maybe" dataset.txt
```

The parameter matrix is printed on the standard output at the end of the execution.

By default, the class association rules concluding on each class must cover every objet of the class. If this requirement is too stringent, fitcare will loosen it by itself. However, this may take much time. Hence, if some classes contain known proportions of noise (or misclassified objects), option --proportions (-p) can be used. It sets minimal proportions of covered objects in each class. On the command line, do not forget to protect the list of proportions by using double quotes ("). In the previous example, if the class "maybe" is known to contain at least 6.25% of junk (or misclassified objects), fitcare can be called as follows:

```
$ fitcare -c "yes no maybe" -p "1 1 0.925" dataset.txt
```

To avoid costly (and possibly fruitless) extractions of very specific set of rules, option --frequencies (-f) constrains fitcare to keep their frequencies, inside their class, above the given thresholds. On the command line, do not forget to protect the list of frequencies by using double quotes ("). In the previous example, if you want any class association rule concluding on "yes" or "no" to match at least 25% of the objects in their respective classes (and do not want such a constraint on the rules concluding on "maybe"), you must call fitcare as follows:

```
$ fitcare -c "yes no maybe" -f "0.25 0.25 0" dataset.txt
```

Option --all (-a) is used to print not only the final set of class association rules (which may overfit the data set) but every good set of rules during the learning process (which goes from general rules to specific ones). The output files are named according to the argument of option --out + an integer (starting from 0).

OUTPUT DATA

Option --out (-o) is used to set the output file name. When omitted, the output file name is the input file name + ".car". For example, if dataset is "dataset.txt", the default output file name is "dataset.txt.car".

The output data is composed of:
* a preamble describing the format ended with a line "End Of Parameters", and
* the rules.

Every rule is stated in one line composed of the list of attributes of the body of the rule followed by the normalized frequencies of the rules in each class (the frequencies in the positive class sum to 1). The string separating the different elements of the rules are listed in the preamble:
* after "classes=", the classes on which the rules are concluding,
* after "ois=", the string separating the attributes of the bodies of the rules,
* after "bfs=", the string separating the bodies of the rules from the normalized frequencies of these rules in each class,
* after "fs=", the string separating the normalized frequencies. * if the classifer includes class association rules with negations of attributes, after "np=", the prefix indicating such a negation.

These separating strings can be chosen:

The attributes are separated by a string set with option --ois (by default " ").

The body of the class association rule is separated from the normalized frequencies with the string set with option --bfs (by default " : ").

The normalized frequencies are separated by a string set with option --fs (by default " ").

PROPORTION OF TESTING OBJECTS

By default, the classifiers are assessed on the learning objects. When learning the thresholds, the data set may be overfitted. Option --testing (-t) is used to specify a proportion of objects that are set appart for testing purpose only (the CARs are not directly derived from these objects). For example, to test the considered classifiers on 25 of the objects, fitcare can be called as follows:

```
$ fitcare -c "yes no maybe" -t 0.25 dataset.txt
```

NEGATION

Option --neg (-n) makes fitcare extract rules which can contain negations of attributes.

VERBOSE MODE

Option --verbose (-v) makes fitcare display (on the standard output) information about every extraction. These information depend on which options were chosen when fitcare was compiled. However, fitcare will, at least, print what is the current extraction and whether the extracted class association rules covers all the objects of the class.

BUGS

Report bugs to <magicbanana@gmail.com>.

SEE ALSO

fitcarc(1)

COPYRIGHT

© 2008 INSA-Lyon Contributor: Loic Cerf (magicbanana@gmail.com)

NAME

Fitcarc Is The Class Association Rule Classifier

SYNOPSIS

fitcarc [options] --rules rules dataset
fitcarc --help | --version

OVERVIEW

From a set of class association rules (typically output by fitcare), fitcarc predicts which class an object of dataset probably belongs to.

RETURN VALUES

fitcare returns values which can be used when called from a script. They are conformed to sysexit.h.

* 0 is returned when fitcarc terminates successfully.
* 64 is returned when fitcarc was called incorrectly.
* 65 is returned when input data is not properly formatted.
* 74 is returned when data could not be read or written on the disk.

GENERAL

* fitcarc --help (or simply fitcarc -h) displays a reminder of the different options that can be passed to fitcarc.
* fitcarc --version (or simply fitcarc -V) displays version information.

OPTIONS

Any option passed to fitcarc may either be specified on the command line or in an option file. If an option is present both in the option file and on the command line, the latter is used.

The option file can be specified through option --opt. When omitted, a file named as the input data file + ".opt" is supposed to be the option file related to the input data file. For example, if dataset is "dataset.txt" and "dataset.txt.opt" exists, it is supposed to be the related option file.

The options have the same name in the option file as on the command line. Only long names can be used though. Arguments passed to an option are separated from the name of the option by "=". For example, these lines may constitute an option file:

```
rules = /home/user/dataset.rules
iis = ","
ois = ","
```

INPUT DATA

The name of the file containing the input data set must be passed as argument dataset.

The syntax of the input data set is flexible. Every line must be either empty (it is ignored) or contain all the attributes of an object. Several different characters may be used to separate the attributes. The attributes can be any string (as far as they do not include any of the separators!). Let us take an example. This line could be part of an input data set:

```
male basketball, gymnastics
```

The object related to this line is described by three attributes: male, basketball and gymnastics. The attributes are separated by the characters " " or/and ",". The fact that two attributes may be separated by several separators does not raise any trouble.

To be properly understood, the user must indicate the separators other wise defaults are used (" "). The related option is --iis for Input Item Separators. In the previous example, fitcarc can be called as follows:

```
$ fitcarc --iis " ," dataset.txt
```

RULES

Option --rules (-r) is mandatory. Its argument rules is a file containing class association rules. It typically is the output of an extraction with fitcare. The manpage of fitcare provides more information about its format.

OUTPUT DATA

Option --out (-o) is used to set the output file name. When omitted, the output file name is the input file name + ".classified". For example, if dataset is "dataset.txt", the default output file name is "dataset.txt.classified".

The output data is composed of:

* a preamble describing the format ended with a line "End Of Parameters", and
* the rules.

Every rule is stated in one line composed of the list of attributes of the body of the rule followed by the normalized frequencies of the rules in each class (the frequencies of all classes sum to 1). The string separating the different elements of the rules are listed in the preamble:

* after "classes=", the classes on which the rules are concluding,
* after "ois=", the string separating the attributes of the objects,
* after "bfs=", the string separating the objects from the normalized frequencies of the class association rules matching them (or the most probable class attribute if --simple is used),
* after "fs=", the string separating the normalized frequencies of the class association rules matching the object.

These separating strings can be chosen:

The attributes are separated by a string set with option --ois (by default " ").

The object is separated from the prediction with the string set with option --bfs (by default " : ").

The normalized frequencies are separated by a string set with option --fs (by default " ").

Option --simple (-s) simplifies the prediction form. Instead of outputting the normalized frequencies of the class association rules matching the object, the most probable class (related to the greatest frequency) is output.

BUGS

Report bugs to <magicbanana@gmail.com>.

SEE ALSO

fitcare(1)

COPYRIGHT

© 2008 INSA-Lyon Contributor: Loïc Cerf (magicbanana@gmail.com)

Bibliographie

[AC06] Bavani Arunasalam and Sanjay Chawla. CCCS : A top-down associative classifier for imbalanced class distribution. In *Proceedings ACM SIGKDD'06*, pages 517–522, 2006.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *Proceedings ACM SIGMOD'93*, pages 207–216, 1993.

[AN07] Arthur Asuncion and David Newman. UCI machine learning repository, 2007. http://archive.ics.uci.edu/ml/.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings VLDB'94*, pages 487–499. Morgan Kaufmann, 1994.

[Bay98] Roberto J. Bayardo. Efficiently mining long patterns from databases. In *ACM SIGMOD'98*, pages 85–93, 1998.

[BB00] Jean-François Boulicaut and Artur Bykowski. Frequent closures as concise representation for binary data mining. In *Proceedings 4th Pacific-Asia conference on Knowledge Discovery and Data mining PAKDD'00*, volume 1805 of *LNCS*, pages 62–73. Springer, 2000.

[BBJ+02] Céline Becquet, Sylvain Blachon, Baptiste Jeudy, Jean-François Boulicaut, and Olivier Gandrillon. Strong association rule mining for large gene expression data analysis : a case study on human SAGE data. *Genome Biology*, 12, 2002.

[BBR00] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Approximation of frequency queries by means of free-sets. In *Proceedings PKDD'00*, volume 1910 of *LNCS*, pages 75–85. Springer, 2000.

[BBR03] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Free-sets : A condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery*, 7(1) :5–22, 2003.

[BC01] Jean-François Boulicaut and Bruno Crémilleux. δ-strong classification rules for characterizing chemical carcinogens. In *Proceedings Predictive Toxicology Challenge co-located with PKDD'01*, 2001.

[BC04] Elena Baralis and Silvia Chiusano. Essential classification rule sets. *ACM Transactions on Database Systems*, 29(4) :635–674, 2004.

[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[BGZ04] Roberto J. Bayardo, Bart Goethals, and Mohammed Javeed Zaki, editors. *IEEE ICDM'04 Workshop on Frequent Itemset Mining Implementations*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[Bir67] Garrett Birkhoff. *Lattice Theory*. American Mathematical Society, 1967.

[BM70] Marc Barbut and Bernard Monjardet. *Ordre et Classification, Algèbre et Combinatoire*. Hachette, 1970.

[BMR02] James Bailey, Thomas Manoukian, and Kotagiri Ramamohanarao. Fast algorithms for mining emerging patterns. In *PKDD'02*, pages 39–50, 2002.

[BMS97] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets : Generalizing association rules to correlations. In *SIGMOD'97*, pages 265–276. ACM Press, 1997.

[BPRB06] Jérémy Besson, Ruggero G. Pensa, Céline Robardet, and Jean-François Boulicaut. Constraint-based mining of fault-tolerant patterns from boolean data. In *KDID'05 Selected and Invited Revised Papers*, volume 3933 of *LNCS*, pages 55–71. Springer, 2006.

[BR01] Artur Bykowski and Christophe Rigotti. A condensed representation to find frequent patterns. In *PODS'01*, 2001.

[BR03] Artur Bykowski and Christophe Rigotti. DBC : a condensed representation of frequent patterns for efficient mining. *Infomation Systems*, 28(8) :949–977, 2003.

[BRB05a] Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Approximation de collections de concepts formels par des bi-ensembles denses et pertinents. In *CAp'05*, pages 313–328, 2005.

[BRB05b] Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Mining formal concepts with a bounded number of exceptions from transactional data. In *Proceedings KDID'04*, volume 3377 of *LNCS*, pages 33–45. Springer, 2005.

[BRB06] Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Mining a new fault-tolerant pattern as an alternative to formal concept discovery. In *Proceedings ICCS'06*, volume 4068 of *LNAI*, pages 144–157. Springer, 2006.

[BRBR05] Jérémy Besson, Céline Robardet, Jean-François Boulicaut, and Sophie Rome. Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis*, 9(1) :59–82, 2005.

[BRM05] Jean-François Boulicaut, Luc De Raedt, and Heikki Mannila, editors. *Constraint-Based Mining and Inductive Databases, European Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany,*

March 11-13, 2004, Revised Selected Papers, volume 3848 of *LNCS*. Springer, 2005.

[BTP+00] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2) :66–75, 2000.

[BU95] Carla E. Brodley and Paul E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1) :45–77, 1995.

[CB91] Peter Clark and Robin Boswell. Rule induction with CN2 : Some recent improvements. In *EWSL'91*, pages 151–163, 1991.

[CB02] Bruno Crémilleux and Jean-François Boulicaut. Simplest rules characterizing classes generated by delta-free sets. In *Proceedings ES'02*, pages 33–46. Springer, 2002.

[CCHJ08] Nitesh V. Chawla, David A. Cieslak, Lawrence O. Hall, and Ajay Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17(2) :225–252, 2008.

[CG02] Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In *PKDD'02*, pages 74–85, 2002.

[CG03] Toon Calders and Bart Goethals. Minimal k-free representations of frequent sets. In *PKDD'03*, pages 71–82, 2003.

[CG07] Toon Calders and Bart Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1) :171–206, 2007.

[CGSB08] Loïc Cerf, Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. A parameter free associative classifier. In *Proceedings DaWaK'08*, volume 5182 of *LNCS*, pages 238–247. Springer, 2008.

[CJK04] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial : special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(1) :1–6, 2004.

[CJZ09] Nitesh V. Chawla, Nathalie Japkowicz, and Zhi-Hua Zhou, editors. *PAKDD'09 Workshop : Data Mining When Classes are Imbalanced and Errors Have Costs*, 2009.

[CL01] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : A library for support vector machines, 2001. http ://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

[CM02] Antoine Cornuéjols and Laurent Miclet. *Apprentissage Artificiel : Concepts et algorithmes*. Eyrolles, 2002.

[Coe04] Frans Coenen. The LUCS-KDD software library, 2004. http ://www.csc.liv.ac.uk/∼frans/KDD/Software/.

[Coh95] William W. Cohen. Fast effective rule induction. In *ICML'95*, pages 115–123, 1995.

[CRB05] Toon Calders, Christophe Rigotti, and Jean-François Boulicaut. A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNCS*, pages 64–80. Springer, 2005.

[CYHH07] Hong Cheng, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings ICDE'07*, pages 716–725. IEEE Computer Society, 2007.

[DL99] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns : discovering trends and differences. In *Proceedings KDD'99*, pages 43–52. ACM Press, 1999.

[DZWL99] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. CAEP : Classification by aggregating emerging patterns. In *Proceedings DS'99*, volume 1721 of *LNCS*, pages 30–42. Springer, 1999.

[EM05] Yasser El-Manzalawy. WLSVM : Integrating libsvm into weka environment, 2005. http ://www.cs.iastate.edu/∼yasser/wlsvm/.

[FI93] Usama M. Fayyad and K. B. Irani. Multi-interval discretization of continous-valued attributes for classification learning. In *Proceedings IJCAI'93*, pages 1022–1027. Morgan Kaufmann, 1993.

[FPSSU96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[FR03] Hongjian Fan and Kotagiri Ramamohanarao. A bayesian approach to use emerging patterns for classification. In *Proceedings of the Fourteenth Australasian Database Conference on Database Technologies*, pages 39–48, Darlinghurst, Australia, 2003. Australian Computer Society, Inc.

[Fre00] Alex Alves Freitas. Understanding the crucial differences between classification and discovery of association rules - a position paper. *SIGKDD Explorations*, 2(1) :65–69, 2000.

[FSFG+10] Frédéric Flouvat, Nazha Selmaoui-Folcher, Dominique Gay, Isabelle Rouet, and Chloe Grison. Constrained colocation mining : application to soil erosion characterization. In *Proceedings SAC'10*. ACM Press, 2010. To appear.

[Für02] Johannes Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2 :721–747, 2002.

[FW94] Johannes Fürnkranz and Gerhard Widmer. Incremental reduced error pruning. In *ICML'94*, pages 70–77, 1994.

[GFF+08] Rohit Gupta, Gang Fang, Blayne Field, Michael Steinbach, and Vipin Kumar. Quantitative evaluation of approximate frequent pattern mining algorithms. In *KDD'08 : Proc. of the 14th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 301–309. ACM Press, 2008.

[GKL06] Gemma C. Garriga, Petra Kralj, and Nada Lavrac. Closed sets for labeled data. In *Proceedings PKDD'06*, pages 163–174. Springer, 2006.

[GKL08] Gemma C. Garriga, Petra Kralj, and Nada Lavrac. Closed sets for labeled data. *Journal of Machine Learning Research*, 9 :559–580, 2008.

[GMT05] Bart Goethals, Juho Muhonen, and Hannu Toivonen. Mining non-derivable association rules. In *SIAM DM'05*, 2005.

[GRM+07] Dominique Gay, Isabelle Rouet, Morgan Mangeas, Nazha Selmaoui, and Pascal Dumas. Assessment of classification methods for soil erosion risks. In *MODSIM'07*, 2007.

[GSB07] Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. Pattern-based decision tree construction. In *Proceedings of the 2nd International Conference on Digital Information Management ICDIM'07*, pages 291–296. IEEE Press, 2007.

[GSB08] Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. Feature construction based on closedness properties is not that simple. In *Proceedings PAKDD'08*, volume 5012 of *LNCS*, pages 112–123. Springer, 2008.

[GSB09] Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. Application-independent feature construction from noisy samples. In *Proceedings PAKDD'09*, volume 5476 of *LNCS*, pages 965–972. Springer, 2009.

[GSW05] Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.

[GZ03] Bart Goethals and Mohammed Javeed Zaki, editors. *IEEE ICDM'03 Workshop on Frequent Itemset Mining Implementations*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[GZ04] Bart Goethals and Mohammed Javeed Zaki. Advances in frequent itemset mining implementations : report on FIMI'03. *SIGKDD Explorations*, 6(1) :109–117, 2004.

[HC06] Céline Hébert and Bruno Crémilleux. Optimized rule mining through a unified framework for interestingness measures. In *Proceedings DaWaK'06*, volume 4081 of *LNCS*, pages 238–247. Springer, 2006.

[HK00] Jiawei Han and Micheline Kamber. *Data Mining : Concepts and Techniques*. Morgan Kaufmann, 2000.

[HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, *ACM SIGMOD'00*, pages 1–12. ACM, 2000.

[JL95] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings UAI'95*, pages 338–345. Morgan Kaufmann, 1995.

[KG02] Marzena Kryszkiewicz and Marcin Gajek. Concise representation of frequent patterns based on generalized disjunction-free generators. In *PAKDD'02*, pages 159–171, 2002.

[KM03] Jeremy Kubica and Andrew W. Moore. Probabilistic noise identification and data cleaning. In *Proceedings ICDM'03*, pages 131–138. IEEE Computer Society, 2003.

[KO02] Sergei O. Kuznetsov and Sergei A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3) :189–216, 2002.

[LDR00a] Jinyan Li, Guozhu Dong, and Kotagiri Ramamohanarao. Instance-based classification by emerging patterns. In *Proceedings The 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 191–200. Springer, 2000.

[LDR00b] Jinyan Li, Guozhu Dong, and Kotagiri Ramamohanarao. Making use of the most expressive jumping emerging patterns for classification. In *Proceedings PAKDD'00*, volume 1805 of *LNCS*, 2000.

[LDR01] Jinyan Li, Guozhu Dong, and Kotagiri Ramamohanarao. Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information Systems*, 3(2) :131–145, 2001.

[LDRW04] Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao, and Limsoon Wong. DeEPs : A new instance-based lazy discovery and classification system. *Machine Learning*, 54(2) :99–124, 2004.

[LHM98] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings KDD'98*, pages 80–86. AAAI Press, 1998.

[LHP01] Wenmin Li, Jiawei Han, and Jian Pei. CMAR : Accurate and efficient classification based on multiple class-association rules. In *Proceedings ICDM'01*, pages 369–376. IEEE Computer Society, 2001.

[LLW07] Jinyan Li, Guimei Liu, and Limsoon Wong. Mining statistically important equivalence classes and delta-discriminative emerging patterns. In *Proceedings of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining KDD'07*. ACM Press, 2007.

[LRD00] Jinyan Li, Kotagiri Ramamohanarao, and Guozhu Dong. The space of jumping emerging patterns and its incremental maintenance algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 551–558, 2000.

[LRD01] Jinyan Li, Kotagiri Ramamohanarao, and Guozhu Dong. Combining the strength of pattern frequency and distance for classification. In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 455–466, 2001.

[MT96] Heikki Mannila and Hannu Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *KDD*. AAAI Press, 1996.

[MT97] Heikki Mannila and Hannu Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3) :241–258, 1997.

[NH05] Canh Hao Nguyen and Tu Bao Ho. An imbalanced data rule learner. In *PKDD'05*, pages 617–624, 2005.

[PB05] Ruggero G. Pensa and Jean-François Boulicaut. From local pattern mining to relevant bi-cluster characterization. In *IDA'05*, volume 3646 of *LNCS*, pages 293–304. Springer, 2005.

[PBTL98] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Pruning closed itemset lattices for associations rules. In *BDA'98*, 1998.

[PBTL99] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1) :25–46, 1999.

[PF07] Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *ECML'07*, pages 658–665, 2007.

[PRB06] Ruggero G. Pensa, Céline Robardet, and Jean-François Boulicaut. Supporting bi-cluster interpretation in 0/1 data by means of local patterns. *Intelligent Data Analysis*, 10(5) :457–472, 2006.

[QCJ93] J. Ross Quinlan and R. Mike Cameron-Jones. FOIL : A midterm report. In *ECML'93*, pages 3–20, 1993.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, 1986.

[Qui93] J. Ross Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann, 1993.

[RB07] Umaa Rebbapragada and Carla E. Brodley. Class noise mitigation through instance weighting. In *Proceedings ECML'07*, volume 4701 of *LNCS*, pages 708–715. Springer, 2007.

[RF07] Kotagiri Ramamohanarao and Hongjian Fan. Patterns based classifiers. *World Wide Web*, 10(1) :71–83, 2007.

[SFFG+09] Nazha Selmaoui-Folcher, Frédéric Flouvat, Chloé Grison, Isabelle Rouet, and Dominique Gay. Découverte de colocations dans un SIG : extension et application à l'étude de l'érosion en nouvelle-calédonie. In *SAGEO'09*, 2009.

[SGB09] Nazha Selmaoui, Dominique Gay, and Jean-François Boulicaut. Construction de descripteurs pour classer à partir d'exemples bruités. In *Actes d'EGC'09*, RNTI-E-15, pages 91–102. CEPADUES, 2009.

[SLGB06] Nazha Selmaoui, Claire Leschi, Dominique Gay, and Jean-François Boulicaut. Feature construction and delta-free sets in 0/1 samples. In *Proceedings DS'06*, volume 4265 of *LNCS*, pages 363–367. Springer, 2006.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[UB90] Paul E. Utgoff and Carla E. Brodley. An incremental method for finding multivariate splits for decision trees. In *ICML'90*, pages 58–65, 1990.

[VC07] Florian Verhein and Sanjay Chawla. Using significant positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proceedings IEEE ICDM'07*, pages 679–684. IEEE Computer Society, 2007.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining : Practical machine learning tools and techniques (2nd edition)*. Morgan Kaufmann, 2005.

[Wil82] Robert Wille. Restructuring lattice theory : an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered stes*, pages 445–470. Reidel, 1982.

[WK05] Jianyong Wang and Georges Karypis. HARMONY : efficiently mining the best rules for classification. In *Proceedings SIAM SDM'05*, pages 34–43, 2005.

[WK06] Jianyong Wang and George Karypis. On mining instance-centric classification rules. *IEEE Transactions on Knowledge and Data Engineering*, 18(11) :1497–1511, 2006.

[WKQ+08] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus F. M. Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1) :1–37, 2008.

[WLW04] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5 :975–1005, 2004.

[YH03] Xiaoxin Yin and Jiawei Han. CPAR : Classification based on predictive association rules. In *Proceedings SIAM SDM'03*, pages 369–376, 2003.

[YWZ04] Ying Yang, Xindong Wu, and Xingquan Zhu. Dealing with predictive-but-unpredictable attributes in noisy data sources. In *Proceedings PKDD'04*, volume 3202 of *LNCS*, pages 471–483. Springer, 2004.

[Zak00] Mohammed J. Zaki. Generating non-redundant association rules. In *Proceedings ACM SIGKDD'00*, pages 34–43. ACM Press, 2000.

[ZDR00] Xiuzhen Zhang, Guozhu Dong, and Kotagiri Ramamohanarao. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *KDD'00*, pages 310–314, 2000.

[Zim08] Albrecht Zimmermann. Ensemble-trees : Leveraging ensemble power inside decision trees. In *Proceedings DS'08*, volume 5255 of *LNCS*, pages 76–87. Springer, 2008.

[ZW04] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise : A quantitative study. *Artificial Intelligence Revue*, 22(3) :177–210, 2004.

[ZW07] Yan Zhang and Xindong Wu. Noise modeling with associative corruption rules. In *Proceedings ICDM'07*, pages 733–738. IEEE Computer Society, 2007.

[ZWC03] Xingquan Zhu, Xindong Wu, and Qijun Chen. Eliminating class noise in large datasets. In *ICML'03*, pages 920–927, 2003.

[ZWZL08] Shichao Zhang, Xindong Wu, Chengqi Zhang, and Jingli Lu. Computing the minimum-support for mining frequent patterns. *Knowledge and Information Systems*, 15(2) :233–257, 2008.